Escuela Politécnica Superior

Sistema para la predicción del rendimiento de los alumnos en el e-learning

Máster Universitario en Ciencia de Datos

Trabajo Fin de Máster

Autor:
Omar Moukket Rkizat
Tutor/es:
David Gil Méndez
Sergio Luján Mora

Universitat d'Alacant
Universidad de Alicante

Septiembre 2021

Todas las figuras y tablas expuestas en este documento son de elaboración propia a no ser que se indique explícitamente la fuente de procedencia.

Resumen

El aprendizaje electrónico o e-learning es una modalidad de enseñanza que ha crecido de manera exponencial en los últimos años. Esto es debido a las grandes ventajas que ofrece, como puede ser la flexibilidad de acceso desde cualquier localización del mundo y a cualquier hora del día, la posibilidad de llegar a un gran número de personas con un aforo ilimitado, y la reducción de grandes costos a empresas e instituciones de educación. Ante la llegada de la pandemia de COVID-19 se ha producido la mayor interrupción de los sistemas educativos jamás producida en la historia, afectando a millones de estudiantes alrededor de todo el mundo. Frente a este escenario, el e-learning ha sido el foco central produciéndose una migración masiva de la enseñanza al marco virtual, evitando que la educación mundial quede bloqueada y dejando más que demostrada la importancia del e-learning.

Sin embargo, este cambio de paradigma no es tan sencillo de llevar a cabo, puesto que se trata de una metodología que pone a los estudiantes en el centro del aprendizaje con implicaciones que van más allá de la traslación de la exposición presencial del docente al marco virtual. La principal diferencia de esta metodología de enseñanza respecto a la tradicional es la no presencialidad del docente y, por lo tanto, la no disponibilidad de una tutorización directa entre el alumno y el profesor. Por ello, hay que aprovechar los avances tecnológicos para crear herramientas que permitan ayudar a reforzar la calidad del e-learning.

En este Trabajo Final de Máster, partiendo de los datos monitorizados en un curso e-learning, el objetivo será crear un sistema predictivo que sea capaz de conocer cuál será el rendimiento de los estudiantes en las pruebas de evaluación de tipo test basándose en el progreso de estos. Mediante esta predicción, por un lado, los estudiantes podrán conocer cuán preparados van de cara a la próxima prueba de evaluación antes de enfrentarse a ella y, por otro lado, los docentes podrán identificar en una etapa temprana aquellos casos de estudiantes que no están alcanzando una correcta evolución en el curso, pudiendo ofrecerles una asistencia personalizada y evitando así los efectos de cualquier barrera de aprendizaje.

Para lograr este objetivo, en primer lugar, se lleva a cabo un proceso de preprocesamiento detallado de los datos para lograr la estructura de datos final necesaria para entrenar los modelos predictivos. En segundo lugar, con el objetivo de aumentar la cantidad de los datos, se lleva a cabo un proceso de generación de datos sintéticos mediante ecuaciones lineales. Por último, se hace uso de diferentes técnicas y enfoques de métodos de aprendizaje automático para lograr la solución final mediante la cual se ha conseguido lograr un rendimiento de predicción muy alto, lo cual hace que el resultado final sea una herramienta muy fiable y útil de cara a la predicción del rendimiento de los usuarios en el e-learning.

Motivación, justificación y objetivo general

A la hora de decidir qué proyecto escoger para llevar a cabo el Trabajo de Fin de Máster (TFM) tenía varias ideas planteadas, además de las disponibles en el listado de propuestas por parte del profesorado. Entre todas ellas, una era continuar con las futuras líneas de trabajo que quedaron planteadas en mi Trabajo de Fin de Grado (TFG) del grado en Ingeniería Informática. Al mismo tiempo, daban comienzo las primeras clases del curso en las cuales me reencontré con uno de mis antiguos tutores del TFG, David Gil Méndez, formando parte del profesorado de una de las asignaturas del máster. En dicho reencuentro conversamos sobre la posibilidad de retomar el trabajo que llevamos a cabo en el grado para dejar el proyecto aún más completo, contando también con la incorporación de Sergio Luján Mora al equipo, quien fue el coordinador de la propuesta de dicho TFG. Por lo tanto, con este reencuentro, definitivamente me quedó claro cuál iba a ser la propuesta sobre la que quería trabajar en este TFM.

En dicho TFG se implementó un curso e-learning bajo dos versiones, una de ellas con determinados problemas de accesibilidad web para las personas con discapacidad y otra totalmente accesible. Ambas versiones se hicieron públicas para que los usuarios pudieran realizar dichos cursos, mientras se monitorizaba de manera automática la actividad de estos y su proceso de aprendizaje dentro de la plataforma. A partir de los datos recogidos en ambas versiones del curso, el objetivo era realizar una comparación y comprobar cómo influyen los problemas de accesibilidad en el rendimiento y aprendizaje de los usuarios con discapacidad.

En este TFM, a partir de los datos del curso e-learning recogidos en el TFG, el objetivo va a ser aplicar distintas técnicas de machine learning o aprendizaje automático sobre ellos con el fin de conseguir un sistema predictivo final en el cual, a partir del transcurso y el comportamiento del usuario en cada bloque del curso, se pueda predecir cuál será su resultado en la prueba o test correspondiente a dicho bloque. Con esto se conseguiría que cada usuario, antes de realizar el test, mediante la predicción del sistema pueda saber si va lo suficientemente preparado o no para la prueba.

Por lo tanto, sabiendo que voy a continuar trabajando sobre un proyecto que yo mismo he llevado a cabo y contando de nuevo con la ayuda de mis dos antiguos tutores, me sentía con la motivación necesaria para llevar a cabo este TFM de manera viable y segura.

Agradecimientos

A mi familia por el apoyo incondicional que me han brindado desde siempre.

A mis amigos por estar siempre ahí.

A mis tutores David Gil Méndez y Sergio Luján Mora por los consejos, orientación y enseñanza que me han brindado hasta esta etapa.

Citas

"Enseñar en la era de internet significa que debemos enseñar las habilidades de mañana desde hoy."

- *Jennifer Fleming*

Índice de contenidos

Índice de figuras

Índice de tablas

Índice de abreviaturas

ANN: Artificial Neuronal Network

BIC: Bayesian Information Criterion

CSV: Comma-Separated Values

CV: Cross Validation

FFNN: Feed-Forward Neural Network

LRS: Learning Record Store

MAE: Mean Absolute Error

MLP: Multilayer Perceptron

MSE: Mean Square Error

RSS: Residual Sum of Squares

SVM: Support Vector Machine

TSS: Total Sum of Squares

VIF: Variance Inflation Factor

xAPI: Experience API

1. Introducción

El aprendizaje electrónico o e-learning [1] es una modalidad de enseñanza y aprendizaje que ha crecido de manera exponencial durante los últimos años [2]. Este crecimiento es debido a las grandes ventajas que ofrece, como puede ser la flexibilidad de acceso desde cualquier localización del mundo y a cualquier hora del día, la posibilidad de llegar a un gran número de personas con un aforo ilimitado y la reducción de grandes costos a empresas e instituciones de enseñanza. Ante la llegada de la pandemia de COVID-19 se ha producido la mayor interrupción de los sistemas educativos jamás producida en la historia, afectando a casi 1.6 mil millones de estudiantes alrededor de todo el mundo [3]–[5]. Frente a este escenario, el e-learning fue el foco central, produciéndose una migración masiva de la enseñanza al marco virtual, evitando que la enseñanza mundial quede colapsada [6]–[8].

Por la tanto, la importancia del e-learning ha quedado más que demostrada e incluso va a ser un sistema de enseñanza que se va a mantener en muchas instituciones una vez acabada la pandemia. No obstante, este cambio de paradigma no es tan sencillo de llevar a cabo puesto que se trata de una metodología que pone a los estudiantes en el centro del aprendizaje, con implicaciones que van más allá de la traslación de la exposición presencial del docente al marco virtual. Esto se ha visto reflejado durante el traslado urgente de la docencia presencial al ámbito virtual, al comienzo de la pandemia, donde se han tenido que aplicar algunas prácticas de emergencia que los expertos en el ámbito han denominado como "docencia remota de emergencia", puesto que a pesar de ser una docencia virtual no tiene por qué equivaler a una docencia totalmente de calidad [9]–[11].

Para ofrecer una enseñanza virtual de calidad hay que tener en cuenta ciertos requisitos fundamentales, debido a que la principal diferencia de esta metodología respecto a la enseñanza tradicional es la no presencialidad del docente y, por lo tanto, la no disponibilidad de una tutorización o seguimiento directo entre el alumno y el profesor. Por ello, además del diseño de un contenido adaptado y coherente con las circunstancias, es muy importante la monitorización de la actividad de los usuarios dentro del curso e-learning para poder analizar su comportamiento y detectar posibles barreras de aprendizaje o puntos de estancamiento. A su vez, este cambio de paradigma obliga a un cambio en la

metodología de evaluación de los alumnos, siendo esta también trasladada al marco virtual. En definitiva, es un cambio necesario pero muy delicado si se quiere asegurar una docencia de calidad y por consiguiente un correcto aprendizaje por parte de los alumnos.

Por lo tanto, aprovechar los avances tecnológicos es crucial para poder desarrollar herramientas y sistemas que permitan mejorar las plataformas e-learning, pudiendo brindar todas las facilidades posibles y necesarias para los alumnos en este cambio de paradigma, repercutiendo así en la calidad de la docencia.

Con el objetivo de mejorar el proceso de aprendizaje de los usuarios y la calidad del e-learning, en este trabajo la idea es explotar los datos que se registran y almacenan dentro de un curso e-learning a partir de la monitorización de la actividad de los usuarios dentro de este. La finalidad es emplear estos datos para la creación de un sistema capaz de predecir mediante aprendizaje automático cuál será el rendimiento del usuario en la próxima prueba o test del bloque del curso en el que se encuentra.

Esto no tal solo sería de gran ayuda para los estudiantes, permitiéndoles medir cómo van de preparados de cara a la próxima prueba antes de enfrentarse a ella, sino que también permitiría a los docentes o instructores identificar en una etapa temprana aquellos casos de estudiantes que no están alcanzando una correcta evolución en el curso, pudiendo ofrecerles una asistencia personalizada para garantizar el mejor proceso de aprendizaje posible y evitando los efectos de cualquier barrera de aprendizaje.

2. Estudio de viabilidad

Antes de comenzar a abordar los principales apartados del trabajo, es conveniente llevar a cabo un estudio de viabilidad de este proyecto para conocer las limitaciones y los riesgos a los que nos podemos enfrentar. Para ello, vamos a hacer uso de la herramienta de análisis de Debilidades, Amenazas, Fortalezas y Oportunidades o DAFO y, por otro lado, haremos un análisis de riesgos planteando posibles planes de contingencia que puedan permitir la finalización de este proyecto con éxito.

2.1. Análisis DAFO

Este análisis consiste en plasmar en una tabla las debilidades, amenazas, fortalezas y oportunidades del proyecto, pudiendo tener a través de estas propiedades una vista general que nos permita trazar una estrategia para afrontar de la mejor manera posible el transcurso del proyecto. Dichas propiedades se clasifican en base a su origen, que puede ser interno o externo, y en base a su carácter que puede ser positivo o negativo.

Como se puede ver en la Figura 1, dentro del apartado de las propiedades de carácter positivo, tenemos las fortalezas y las oportunidades. En cuanto a las fortalezas, encontramos en primer lugar la motivación y las ganas de aprender, lo cual es algo fundamental a la hora de llevar a cabo cualquier proyecto académico o profesional. En segundo lugar, encontramos como fortaleza el hecho de continuar trabajando sobre un proyecto personal, lo cual significa que partimos de un contexto en el que se tiene experiencia y conocimiento, además del hecho de volver a formar equipo con los mismos tutores del primer proyecto. La tercera fortaleza es la disposición de todos los recursos y herramientas necesarias para trabajar, tanto tecnológicas como pedagógicas, y en último lugar se destaca la nula inversión económica necesaria para el desarrollo de este proyecto.

	Positivos	Negativos
Origen interno	<u>Fortalezas</u> • Motivación y ganas de aprender • Continuar trabajando sobre un proyecto personal • Disposición de recursos y herramientas necesarias para trabajar • Ningún coste económico	<u>Debilidades</u> • Tiempo ajustado para compaginar el proyecto con cuatro asignaturas en paralelo
Origen externo	<u>Oportunidades</u> • Sector en pleno crecimiento y evolución • Afecta e influye en un gran número de usuarios	<u>Amenazas</u> • Aumento de la competencia • No disponer de la cantidad suficiente de datos para modelar el sistema predictivo

Figura 1. Matriz DAFO

En cuanto a las oportunidades, se han destacado dos en concreto. La primera de ellas es el hecho de estar ante un sector que se encuentra en pleno crecimiento y en continua evolución, en el cual se buscan herramientas y sistemas que ayuden a la mejora del e-learning. La segunda oportunidad es el hecho de poder trabajar en la solución a un problema que afecta a un gran número usuarios.

Dentro del apartado de las propiedades de carácter negativo, tenemos las debilidades y las amenazas. En las debilidades se ha destacado el hecho de disponer de un tiempo ajustado para llevar a cabo el desarrollo de este proyecto en paralelo a las cuatro asignaturas del segundo cuatrimestre del curso. Finalmente, se han destacado dos amenazas, en primer lugar, la posibilidad del crecimiento de la competencia debido al gran auge del sector, con la consiguiente aparición de posibles soluciones similares. En segundo lugar, existe la posibilidad de que se tengan que incorporar datos sintéticos para complementar los datos recogidos sobre los que se va a trabajar para el desarrollo de un sistema predictivo con un rendimiento adecuado.

2.2. Análisis de riesgos

Mediante este análisis vamos a exponer una serie de riesgos que creamos que pueden comprometer el desarrollo del proyecto, alterando el ritmo de trabajo o la planificación. Este tipo de análisis es complementario al análisis DAFO, y nos va a permitir trazar una estrategia de contingencia para cada riesgo que expongamos antes de comenzar a trabajar en el desarrollo del proyecto. De esta manera, en caso de producirse en algún momento alguno de estos riesgos, podremos estar preparados para contrarrestar sus efectos. A continuación, se listan los posibles riesgos a los cuales se puede enfrentar este proyecto:

- Enfermar: este es un riesgo que puede ocurrir en cualquier proyecto de manera imprevista, y si tenemos en cuenta que en el momento del desarrollo de este proyecto se sigue en la lucha contra la pandemia de COVID-19 y la población aún no está totalmente inmunizada, este riesgo es aún más grande. Hay dos posibles planes de contingencia ante este riesgo en función de la gravedad de la enfermedad. Teniendo en cuenta que se trata de un trabajo con un peso de 6 créditos (unas 150 horas de trabajo aproximadamente), sería interesante reservar en la planificación una pequeña cantidad de horas que puedan ser empleadas en el caso de un grado de enfermedad leve, pudiendo reducir la carga de trabajo sin comprometer el desarrollo del proyecto, aprovechando las horas reservadas. Sin embargo, si se trata de un grado de enfermedad alto o grave, dichas horas de reserva no serían suficientes para contrarrestar los efectos y el plan de contingencia en este escenario consistiría en presentar el proyecto en la convocatoria extraordinaria.
- Aumento de la carga de trabajo de otras asignaturas: este trabajo se va a llevar a cabo junto a cuatro asignaturas en paralelo, y en algún momento puede ocurrir que alguna o algunas de las asignaturas demanden más carga de trabajo de la esperada. Esto causaría un desequilibrio en el reparto de horas de trabajo de este cuatrimestre, pudiendo afectar al desarrollo de este proyecto. Ante este riesgo, el plan de contingencia sería similar al caso de enfermar levemente, habría que tener un pequeño porcentaje de horas reservadas en la planificación del proyecto para emplearlas si se da esta situación.

- Fallo del equipo informático de trabajo: la rotura de un componente informático del equipo de trabajo se puede producir de manera impredecible, y si se produce en un componente principal se podría dar el caso de una imposibilidad de uso absoluta del equipo. Por otro lado, en función del componente, se puede producir una pérdida del trabajo desarrollado hasta dicho momento, con imposibilidad de recuperarlo. Para lidiar con este riesgo, el primer plan de contingencia va a ser realizar copias de seguridad periódicas, tanto físicas como remotas, sobre el trabajo desarrollado. En caso de un fallo que impida el uso del equipo, el plan de contingencia consistiría en hacer uso de un equipo auxiliar (de familiares, amigos, etc.) mientras se lleva a cabo la reparación del equipo personal.
- Datos monitorizados insuficientes: como se ha adelantado en los puntos introductorios, en este trabajo la idea es explotar los datos monitorizados en un curso e-learning para la implementación de un sistema de predicción. No obstante, estos sistemas necesitan una cantidad mínima de datos para llevar a cabo un proceso de aprendizaje automático adecuado. Por lo tanto, si se da esta situación, el plan de contingencia consistiría en hacer uso de técnicas de generación de datos sintéticos a partir de las características del entorno de los datos reales para complementar a éstos últimos.

Una vez completado el análisis DAFO y el análisis de riesgos, podemos concluir que en el primer caso las propiedades de carácter positivo (fortalezas y oportunidades) son más predominantes que las propiedades de carácter negativo (debilidades y amenazas). En el segundo análisis se han expuesto una serie de riesgos que se pueden producir durante el transcurso del proyecto. No obstante, para cada uno de ellos se han planteado uno o más planes de contingencia fiables que mitigarían o prácticamente anularían los efectos de dichos riesgos. Por lo tanto, podemos concluir que el proyecto es altamente viable, y estamos preparados para implicarnos en las siguientes fases de trabajo.

3. Planificación

Planificar la línea temporal por la que van a pasar los diferentes puntos de trabajo del proyecto es algo fundamental por varios motivos. Nos permite tener una organización personal del proyecto sabiendo en cada momento qué es lo que tenemos que hacer y nos hace estimar un reparto de las horas de trabajo totales entre las diferentes fases, de manera que dediquemos las horas necesarias a cada fase en proporción a su peso en el resultado final del proyecto.

Sabiendo que este trabajo tiene un peso de 6 créditos, con unas 150 horas de trabajo estimadas, y que el comienzo del segundo cuatrimestre es en febrero, vamos a seguir la planificación reflejada en la Tabla 1. En dicha tabla se ha dividido la planificación en tres bloques, cada uno de ellos con una duración proporcional al peso del trabajo que conlleva. Sin embargo, a pesar de tener el reparto dividido por bloques y con fechas de finalización declaradas, esta planificación debe tomarse como herramienta orientativa y no como una guía rígida ya que es posible que alguno o algunos de los apartados del primer o segundo bloque queden completamente finalizados hasta el tercer bloque, donde se completa su desarrollo y redacción en el documento de la memoria del trabajo.

Al primer bloque se le ha asignado un tiempo total de 30 horas, con fecha límite el día 1 de marzo. Puesto que la selección de la idea sobre la que se va a trabajar en este proyecto fue tomada a inicios del mes de febrero, el trabajo de este bloque se reparte durante todo este mes. Se engloban todos los puntos correspondientes a la fase inicial del proyecto: motivación, justificación y descripción del objetivo general, estudio de viabilidad, planificación, estudio del estado de la cuestión, descripción de los objetivos y de la metodología que se va a seguir.

Una vez finalizada la parte introductoria o inicial del proyecto, se pasa al segundo bloque al cual se le ha asignado un tiempo total de 80 horas con fecha límite el día 30 de mayo. En este caso se engloba todo el proceso que va a conllevar el desarrollo y aplicación de diferentes modelos predictivos, con el objetivo de encontrar la mejor solución para nuestro problema. Para ello, primero se tendrán que preprocesar, analizar y preparar los datos sobre los que se va a basar el resto del trabajo.

Tabla 1. Planificación temporal del TFM

Contenidos	Tiempo total	Fecha límite de finalización
Motivación, justificación y objetivo general Estudio de viabilidad Planificación Estado de la cuestión Objetivos Metodología	30 horas	1 de marzo
Análisis y tratamiento de los datos Desarrollo del sistema predictivo	80 horas	30 de mayo
Resultados Conclusiones y trabajo futuro Resumen Introducción Referencias y bibliografía Agradecimientos y citas Finalización y revisión de la memoria	40 horas	4 de junio

Por último, el tercer bloque recibe un total de 40 horas, con fecha límite el 4 de junio. En esta fase final se abordarán los siguientes apartados: resultados de la solución final, conclusiones y trabajo futuro, introducción, resumen, referencias, agradecimientos y citas y, para terminar, se llevará a cabo la revisión y finalización de la redacción de la memoria del trabajo.

4. Estado de la cuestión

Antes de embarcarnos en el desarrollo de nuestra solución, es de gran importancia conocer cuál es el estado de la cuestión acerca del problema que queremos abordar. De esta manera, podremos nutrirnos de conocimiento acerca de las soluciones actuales y de los recursos disponibles.

4.1. Antecedentes

Como se ha adelantado en el apartado de la introducción, este TFM parte de la base del TFG que he desarrollado en el grado de Ingeniería Informática [12]. En dicho trabajo el objetivo principal era desarrollar un curso e-learning sobre una temática concreta, con la finalidad de que este sea publicado de manera online y realizado por usuarios reales. La aplicación web de dicho curso e-learning permitía monitorizar la actividad de los usuarios con un alto nivel de detalle, y dichos datos fueron empleados para realizar un proceso de análisis que permitía detectar barreras de aprendizaje que se pudieran generar dentro del curso debido al diseño pedagógico de este o a los problemas de accesibilidad a los cuales se enfrentan los usuarios con discapacidad.

Para cumplir con este objetivo general, se plantearon subobjetivos que se desarrollaron siguiendo una metodología secuencial. En primer lugar, se hizo un estudio sobre los fundamentos y detalles de la accesibilidad web, para conocer cómo puede afectar a las personas con discapacidad dentro de un curso e-learning. Una vez adquirido el conocimiento necesario sobre la accesibilidad web, el siguiente paso era determinar la temática sobre la cual iba a tratar el curso e-learning a implementar.

La temática escogida fue el "consumismo y cambio climático" (Figura 2), considerando que podría ser una buena idea desarrollar un curso e-learning que permita concienciar a los usuarios que lo realicen acerca de la situación actual del cambio climático provocada por el

Figura 2. Curso e-learning desarrollado en el TFG

nivel de consumismo que llevamos a cabo los ciudadanos día a día. Además, se proporciona una serie de consejos aplicables a la rutina diaria y que permiten mitigar la evolución del cambio climático al mismo tiempo que permiten beneficiarse de un ahorro económico. Por lo tanto, el siguiente paso consistía en escoger el contenido pedagógico en base a esta temática y diseñar la estructura del curso e-learning. Esta estructura estaba formada por tres bloques de dos, nueve y ocho lecciones respectivamente.

El primer bloque es una introducción al curso, el segundo bloque versa acerca de la situación actual del cambio climático, las energías renovables y los mitos acerca de estas, y finalmente se exponen consejos que permiten ahorrar energía. Por último, el tercer bloque trata sobre el consumismo y propone diferentes formas de llevar a cabo un mejor consumo en diferentes aspectos de la vida cotidiana (alimentación, transporte, productos electrónicos, pescado, etc.) ayudando al medio ambiente.

El curso tiene en total tres tests o pruebas de evaluación, dos en el segundo bloque y una en el tercer bloque, cuyos resultados también son monitorizados por el sistema (ejemplo del Test 1 en la Figura 3). Una vez completada la estructura del curso junto a su contenido,

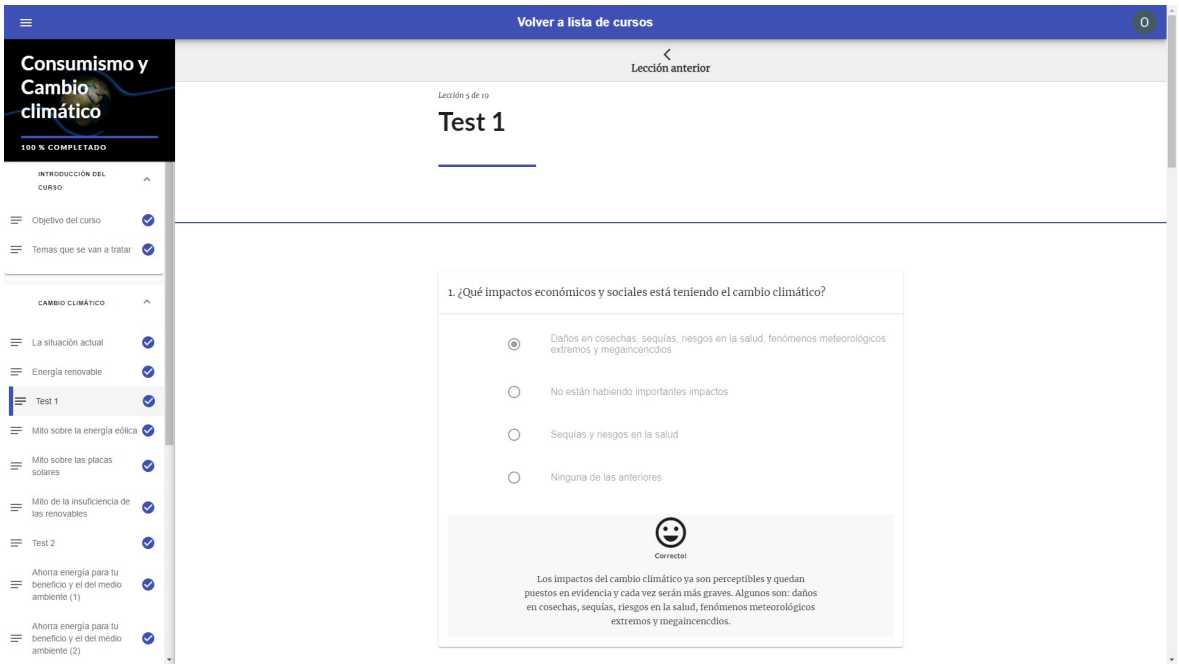

Figura 3. Test 1 del curso e-learning

este se implementó en una aplicación web para hacerlo público como curso e-learning. Para conocer cómo afectan los problemas de accesibilidad web en el e-learning, se tomó la decisión de implementar dos versiones del curso; una versión totalmente accesible[1] y otra versión[2] con el mismo contenido, pero con ciertos problemas de accesibilidad introducidos estratégicamente. Los problemas de accesibilidad introducidos en la segunda versión fueron los siguientes:

- Video: son uno de los principales problemas que se encuentran dentro de la Web las personas con discapacidad. El uso de videos en los que no se proporcionan subtítulos es una directa barrera de aprendizaje para los usuarios que padecen de sordera. Por otro lado, no es suficiente con proporcionar subtítulos automáticos, puesto que estos generan malas subtitulaciones muy frecuentemente, lo cual se ve acentuado si el video contiene ruido de fondo o una pronunciación no muy clara. Otra parte que juega un papel fundamental es el reproductor que aloja a dicho video, el cual debe ser totalmente accesible. Para ello, debe permitir controlar todas las acciones posibles mediante teclado (pausa, reproducción, retroceso, adelantamiento, volumen, mostrar/ocultar subtítulos y emplear la pantalla

[1] https://e-gular.firebaseapp.com
[2] https://e-gulartwo.firebaseapp.com

completa), proporcionar transcripción textual del contenido y ser visible para los lectores de pantalla, haciendo de esta manera su contenido accesible a los usuarios invidentes.

Una vez conocidas todas las características necesarias para que un video sea totalmente accesible en la Web, en la segunda versión del curso se han privado de estas características a los videos que forman parte del contenido didáctico.

- Imágenes: son muy útiles para transmitir o reforzar el contenido didáctico de manera gráfica. Sin embargo, si no se emplean de la manera adecuada son un gran problema para los usuarios invidentes. La forma de hacerlas accesibles consiste en emplear un texto alternativo que describa de manera precisa el contenido que trata de transmitir la imagen en cuestión, de manera que los lectores de pantalla puedan detectar este contenido y transmitirlo a los usuarios invidentes.

  Para cumplir con el objetivo de la segunda versión del curso e-learning, se han retirado los textos alternativos de todas las imágenes empleadas en el contenido didáctico.
- Contenido dinámico: este tipo de contenido varía su visibilidad en función de la interacción del usuario con él. Este tipo de contenido es muy problemático para los usuarios invidentes o con problemas de psicomotricidad, y por ello en la segunda versión del curso se ha introducido una tarjeta dinámica de contenido. Dicha tarjeta tiene dos caras, donde la cara frontal es una imagen cuyo dorso contiene el contenido didáctico. Dicho contenido se hace visible al situar el cursor del ratón encima de esta imagen provocando un movimiento dinámico que rota la tarjeta.

En conclusión, se implementan dos versiones del curso e-learning con el mismo contenido didáctico, pero con la diferencia de que en ciertos puntos de la segunda versión el contenido no es totalmente accesible, mientras sí lo es en la primera. Para conocer la influencia de estos problemas de accesibilidad en el rendimiento del aprendizaje de los usuarios, las pruebas o tests de cada bloque incorporan ciertas cuestiones que evalúan el aprendizaje adquirido justamente a través del contenido que se encuentra en los apartados del curso con problemas de accesibilidad. De esta manera, en el proceso de análisis se ha podido comprobar la diferencia de rendimiento que obtienen los usuarios con discapacidad cuando se encuentran ante un curso e-learning con contenido totalmente accesible y viceversa.

El API que se encarga de monitorizar toda la actividad de los usuarios se llama Experience API (xAPI)[3], y se encarga de generar eventos o statements de diferentes tipos en función del tipo de actividad monitorizada. Estos eventos son almacenados en un contenedor de datos de aprendizaje o Learning Record Store (LRS), siendo empleado en este caso Scorm Cloud[4] debido a su compatibilidad con xAPI. Para cada usuario, los eventos monitorizados y los nombres de cada uno de ellos son los siguientes:

- *Ha empezado*: se entra en el curso e-learning
- *Ha avanzado a*: se pulsa el botón que navega a la siguiente lección
- *Ha vuelto a*: se pulsa en el botón que navega a la anterior lección
- *Ha navegado a*: se navega a una lección en concreto desde el menú
- *Ha observado*: se visualiza un chunk o bloque de contenido por primera vez
- *Ha revisado*: se visualiza un chunk o bloque de contenido que ya había sido visto con anterioridad
- *Ha contestado correctamente/incorrectamente*: se contesta a la cuestión de un test de manera correcta o incorrecta.

A partir de este punto ya conocemos a rasgos generales en qué consiste el proyecto que comprende el punto de partida de este TFM. Concretamente, el punto de partida van a ser los datos que se han podido monitorizar en el curso e-learning gracias a la actividad de los usuarios, con el objetivo de entrenar un modelo predictivo basado en el progreso de estos.

4.2. Aprendizaje automático aplicado al e-learning

A diferencia del enfoque clásico, el e-learning no requiere de la presencia física del docente o de los estudiantes en un aula, por lo que es más flexible, menos costoso y permite la gestión de un mayor número de estudiantes [13]. Por otro lado, la naturaleza de los cursos e-learning ha atraído a un número significante de estudiantes adultos que buscan combinar la educación superior y la formación técnica con sus altas responsabilidades y exigencias

[3] https://xapi.com/
[4] https://rusticisoftware.com/products/scorm-cloud/

laborales [14]. Sin embargo, la alta flexibilidad y expansión del e-learning ha traído consigo grandes retos.

Por ello, muchos estudios científicos [15]–[21] se han enfocado en encontrar la forma de mejorar la calidad de estos cursos e-learning mediante la introducción de innovadoras herramientas y métodos, como la predicción del rendimiento de los usuarios, adaptabilidad automática de la dificultad en función del rendimiento del estudiante, clasificación de los estudiantes en grupos homogéneos, etc.

Junemann, Lagos y Arriagada [22] usaron redes neuronales para predecir el rendimiento escolar que podían tener los estudiantes de 15 años en asignaturas de lectura, matemáticas y ciencias, basándose en sus características familiares, sociales y patrimoniales. Wang y Mitrovic [23] emplearon redes neuronales para estimar el número de errores que puede cometer un alumno al enfrentarse a un problema, basándose en los atributos específicos del problema y las capacidades del alumno. Esta predicción fue aplicada en un único examen para optimizar la selección de problemas que el alumno tiene que resolver en el último proceso de examinación.

Por otro lado, Cripps [24] realizó un estudio basado en datos de estudiantes universitarios. Concretamente, se estudiaron varias características demográficas como la edad, género y raza, así como los resultados de los exámenes de acceso a la universidad para entrenar una red neuronal encargada de estimar si el estudiante va a ser capaz de finalizar el grado universitario junto a su correspondiente nota final. Buenaño, Luján-Mora y Gil [25] aplicaron técnicas de aprendizaje automático para predecir las notas finales de estudiantes universitarios del grado en Ingeniería Informática de Ecuador, basándose en el histórico de datos correspondientes a las notas obtenidas por los estudiantes en 68 asignaturas del grado y empleando árboles de decisión. Por otro lado, Moscoso, Saa y Luján-Mora [26] analizaron datos de estudiantes para predecir la tasa de graduación a partir de las características de los estudiantes matriculados, de manera que se puedan tomar acciones correctivas en una etapa temprana mejorando así el proceso de admisión.

Sheel, Vrooman, Renner y Dawsey [27] compararon el uso de redes neuronales y modelos estadísticos tradicionales para clasificar a los estudiantes en dos grupos, en función de los resultados obtenidos en una única prueba de nivel matemático. Tanto Kalles y Pierrakeas

[28] como Kotsiantis, Pierrakeas y Pintelas [29] han hecho uso de datos derivados de la educación a distancia para predecir el nivel de éxito o fracaso de los exámenes finales mediante diferentes técnicas, entre las cuales están las redes neuronales. Los datos contenían características demográficas, calificaciones de las tareas individuales y el nivel de asistencia a las clases virtuales.

También hay métodos que se han aplicado sobre los resultados obtenidos en pruebas de tipo test, considerando que esta forma de evaluación a diferencia de las pruebas escritas tiene una corrección inequívoca y objetiva según lo establecido por Haladyna [30], minimizando así los factores externos que podrían influir en la calidad de los datos. Siguiendo esta tendencia, Lykourentzou, Giannoukos, Mpardis, Nikolopoulos y Loumos [31], para proponer su solución, han hecho uso de un curso e-learning de nivel introductorio basado en la temática de las redes y telecomunicaciones, publicado en la plataforma Moodle. El curso tenía una duración de 10 semanas y se recogieron los datos de 2006 y 2007, donde se matricularon 37 y 28 alumnos respectivamente, de los cuales completaron el curso 32 y 25 alumnos respectivamente. Gracias a la invariabilidad del contenido del curso durante los diferentes años, se pudo proponer un método que aprovechaba bajo la misma solución todos los datos, permitiendo generar a su vez predicciones para futuros años.

Dicho curso tenía cuatro pruebas o tests de 20 preguntas distribuidos por los diferentes bloques del contenido y un último test global de 40 preguntas. En este experimento se hizo uso de tres Feed-Forward Neural Networks (FFNN) o Redes Neuronales de Alimentación hacia Delante para predecir los resultados de los estudiantes en el último test global a partir de la aproximación de los resultados de los test parciales anteriores.

La implementación fue llevada a cabo mediante MATLAB 2007b, y los pasos que siguieron fueron los siguientes; en primer lugar, se extrajeron los datos de la base de datos del curso e-learning y contando con que el objetivo era predecir los resultados del test final, solo se podían aprovechar los datos de los 57 alumnos que completaron el curso en su totalidad. Este conjunto de datos se dividió en tres subconjuntos de entrenamiento, validación y test respectivamente. El conjunto de entrenamiento contenía el 85% de los datos de las notas de los estudiantes del primer año del curso, lo cual corresponde a las notas de 27 alumnos seleccionados aleatoriamente. El otro 15% (notas de 5 estudiantes) pertenecía al conjunto

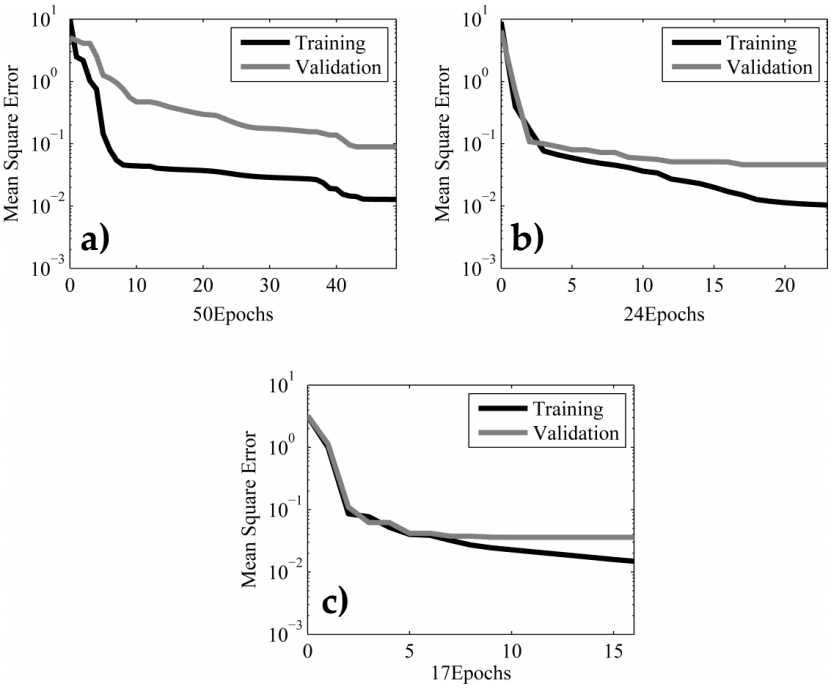

Figura 4 (a)-(c). Rendimiento sobre los datos de entrenamiento y validación de las tres redes FFNN entrenadas durante 17, 24 y 50 epochs
Fuente: [31]

de validación y, por último, el conjunto de datos de test estaba formado por las notas de los 25 estudiantes que completaron el curso en el segundo año.

El hecho de emplear los datos de un año para el proceso de entrenamiento y los de otro año diferente para el proceso de test permitía comprobar si la red es capaz de obtener un buen rendimiento de predicción en base a datos de estudiantes que han pertenecido a semestres diferentes. El siguiente paso fue llevar a cabo el proceso de entrenamiento, y para ello se entrenaron tres redes FFNN que recibían como entrada las notas de los dos, tres y cuatro tests parciales respectivamente y predecían la nota del examen final. Para cada red se llevaron a cabo 100 iteraciones de entrenamiento, escogiendo finalmente la que menor Error Medio Absoluto o Mean Square Error (MAE) obtenía.

Como se puede ver en la Figura 4, donde el eje de ordenadas representa el valor del MAE y el eje de abscisas el número de épocas requeridas para finalizar el proceso de entrenamiento, se obtiene una buena minimización del error entre la predicción de la nota final y la nota final real. El siguiente paso consistió en probar el rendimiento de estas redes sobre el conjunto de datos de test para comprobar la capacidad de generalización obtenida a partir del proceso de entrenamiento. Teniendo en cuenta que las notas del curso están en una escala de 0 a 20, la Figura 5 representa dicho rendimiento para cada una de las redes.

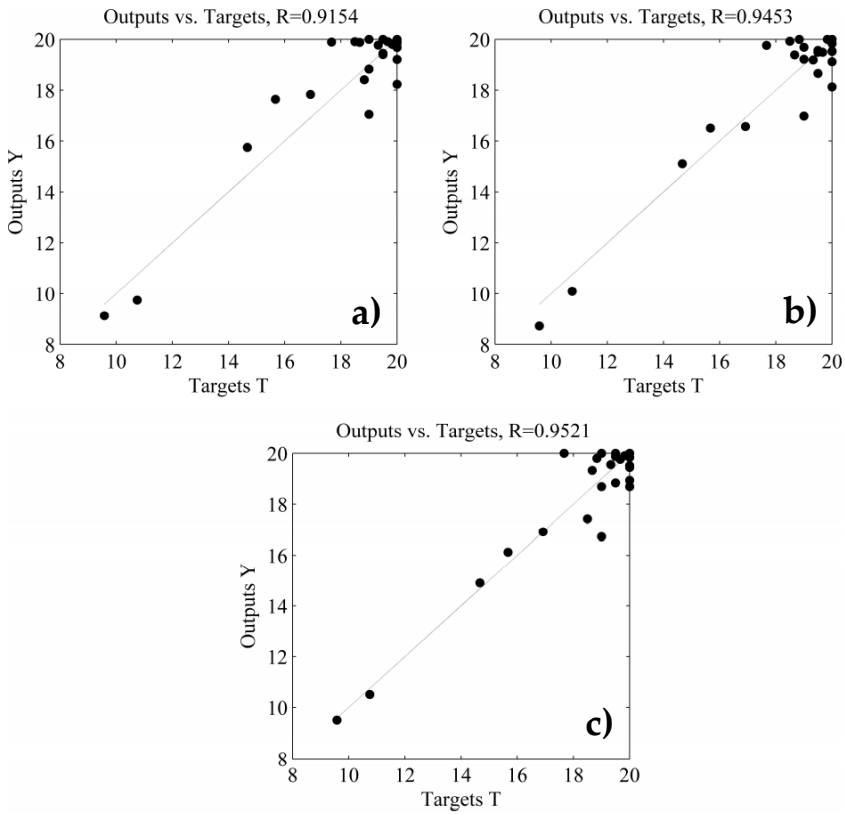

Figura 5 (a)-(c). Resultados de predicción sobre los datos de test mediante las redes neuronales entrenadas
Fuente: [31]

En esta figura el eje de ordenadas refleja la nota predicha y el eje de abscisas la nota real. Por lo tanto, cuanto más cerca esté de la diagonal una coordenada más precisa habrá sido la predicción. La gráfica *a)* (primera red) ya muestra una buena aproximación de las predicciones, lo cual significa que a partir de los resultados de los dos primeros tests y, por lo tanto, a partir de la tercera semana del curso ya se puede tener una alta correlación con el resultado del examen final. Sin embargo, las siguientes gráficas correspondientes a la segunda y tercera red, muestran una ligera mejora respecto a la primera, y la tercera respecto a la segunda. Esto significa que la incorporación progresiva de los datos del tercer y cuarto test también ayudan a mejorar la predicción de la nota final, siendo la incorporación del tercer test más significativa que la del cuarto si observamos el incremento del coeficiente *R*.

Tras comprobar el buen rendimiento obtenido con las redes neuronales se decidió hacer uso de modelos de regresión lineal para comprobar si se obtenía un rendimiento diferente. Para ello se siguió la misma estrategia, ajustando tres modelos lineales que tienen en cuenta los primeros dos, tres y cuatro tests como variables de regresión y la nota del examen final

como variable a predecir. Tras hacer las predicciones sobre el conjunto de datos de test con los modelos lineales, se pudo comprobar que las redes neuronales obtenían mejores resultados, es decir, obtenían un menor MAE.

A su vez, a medida que se aumentaba el número de tests tenidos en cuenta de cara a la predicción, el modelo de regresión lineal disminuía su correlación con la variable respuesta, mientras ocurría lo contrario con la red neuronal. Esto quedó reflejado a través de los valores del coeficiente *R*, el cual en las redes neuronales obtuvo los valores 0.9154, 0.9453 y 0.9521 durante el incremento de tests tenidos en cuenta, mientras que en el modelo de regresión lineal se obtuvieron los valores 0.8100, 0.7613 y 0.7691 respectivamente. Por lo tanto, esta diferencia de rendimiento en el MAE y la bajada progresiva del coeficiente *R* en los modelos de regresión lineal refleja que las redes neuronales han permitido un mejor ajuste a las características no lineales de los datos de este caso de estudio.

Una vez alcanzada la capacidad de predecir el rendimiento de los usuarios a partir de los resultados obtenidos en los tests parciales, los resultados de predicción fueron empleados para agrupar de manera dinámica a los estudiantes con habilidades parejas, con el fin de poder identificar y conocer por parte del docente los requerimientos y necesidades de cada grupo en particular.

Tal como establecieron Feldhusen y Moon [32], agrupar a los estudiantes en función de sus habilidades es esencial para ayudarles a alcanzar su rendimiento académico óptimo y poder mejorar aún más su motivación por aprender. Por lo contrario, no hacer agrupaciones homogéneas puede provocar una bajada en los resultados y en la motivación de los estudiantes. Por lo tanto, tan pronto como se tenga lugar un agrupamiento homogéneo se podrán tomar acciones para promover la mejora del proceso de aprendizaje [33].

En el estudio de predicción del rendimiento que estamos analizando [31], se tienen en cuenta dos grupos diferentes (A y B) basados en los resultados del examen final del curso e-learning, y se ha tenido en cuenta un umbral de clasificación *T* con valor 18.5 dentro de la escala de las notas del curso (de 0 a 20). Se ha establecido este umbral porque el curso es de nivel introductorio y se ha considerado que iba a ser relativamente sencillo para los estudiantes. Por lo tanto, un estudiante pertenecerá al grupo A si obtiene en el examen final una nota superior al umbral *T*, y si es inferior pertenecerá al grupo B. Este proceso de

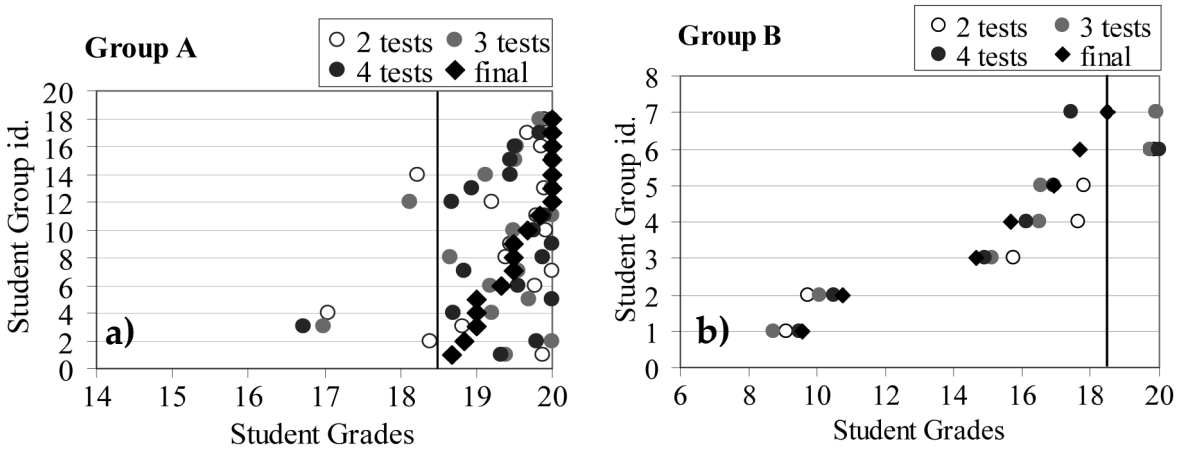

Figura 6 (a)-(b). Clasificación grupal de los estudiantes mediante los resultados de las redes neuronales
Fuente: [31]

clasificación se ha llevado a cabo a través de los resultados predichos por las redes neuronales y los métodos de regresión lineal, y al igual que ocurría en la predicción de la nota final, las redes neuronales han obtenido un mejor rendimiento de clasificación respecto a los métodos de regresión lineal, los cuales proporcionaron clasificaciones menos homogéneas.

Es de vital importancia conseguir grupos homogéneos, puesto que por lo contrario se estaría proporcionando menos asistencia a los alumnos que la requieran o se estaría obstaculizando el progreso de estudiantes que no necesitan dicha asistencia, además de sobrecargar al docente.

La Figura 6 contiene los resultados de clasificación obtenidos mediante la predicción de las redes neuronales y la Figura 7 corresponde a los resultados de clasificación obtenidos mediante la predicción de los métodos de regresión lineal. En estas gráficas se muestra para cada alumno la clasificación grupal en función del nivel de tests tenidos en cuenta (2, 3 y 4) y la nota final. Como se puede ver, en cada gráfica hay una línea vertical divisoria en el valor 18.5 que representa el umbral *T* de clasificación. En las figuras 6 (a) y 7 (a) la situación de los puntos a la derecha de la línea divisoria refleja una correcta clasificación en el grupo A, y si se sitúan a la izquierda significa una clasificación incorrecta en este grupo.

Por otro lado, las figuras 6 (b) y 7 (b) reflejan la misma información, pero para la clasificación del grupo B, donde la situación de los puntos a la izquierda de la línea divisoria significa la correcta clasificación. Por lo general se puede observar que las redes neuronales han obtenido mejores resultados de clasificación en comparación a los métodos de

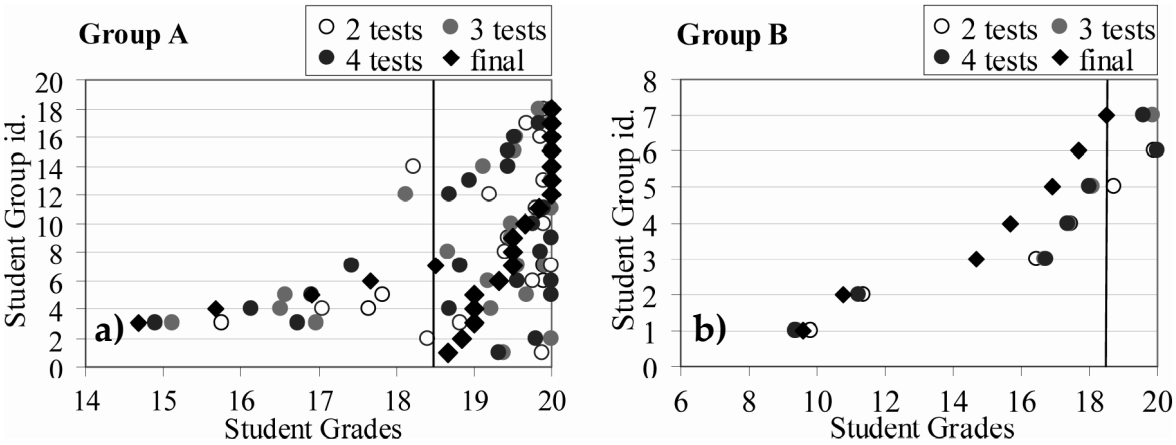

Figura 7 (a)-(b). Clasificación grupal de los estudiantes mediante los resultados de los métodos de regresión lineal
Fuente: [31]

regresión lineal, sobre todo en la clasificación del grupo A donde las redes neuronales tienen una significante menor tasa de error.

Según el análisis de clasificación de estudiantes de este estudio, se puede llegar a la conclusión de que es posible llevar a cabo una agrupación de estudiantes en una época temprana del curso e-learning, más específicamente en la tercera semana del curso como ocurre en este caso. Mediante esta información, los docentes pueden determinar cuáles son las acciones más apropiadas para proporcionar a cada grupo una asistencia adicional adaptada a sus necesidades.

Los cursos e-learning, a pesar de atraer a un gran número de estudiantes, muestran una tasa de abandono mayor que la obtenida con la enseñanza tradicional, superándola en muchos casos por un 10-20% [34]. Otros estudios también confirman este hecho, reflejando unas tasas de abandono en el e-learning del 25% al 40% en comparación a la docencia presencial que suele tener una tasa de abandono con valores entre el 10% y 20% [35]–[38].

Las tasas de retención de estudiantes son uno de los indicadores que las universidades, educadores, responsables políticos y organismos de financiación de la educación superior consideran como medida objetiva del reflejo de la calidad que ofrece una institución educativa. Por ello, este indicador de calidad ha sido reconocido y utilizado a nivel internacional, incluidos Australia, Unión Europea, Estados Unidos y Sudáfrica [39].

Este crecimiento del énfasis por la retención, combinado con las altas tasas de abandono que presentan los cursos e-learning, hace que la reducción de las tasas de abandono sea un aspecto fundamental para la aceptación y éxito de este tipo de cursos. Uno de los aspectos

fundamentales para reducir la posibilidad de abandono en los cursos e-learning es la identificación precisa y rápida de los estudiantes que puedan estar ante este riesgo.

Tan pronto como se pueda identificar a este tipo de estudiantes, los docentes o instructores podrán abordar mejor las necesidades específicas de estos estudiantes y tomar las acciones apropiadas para reducir la probabilidad de abandono en el curso e-learning en cuestión. Por ello, en los últimos años muchos estudios [40]–[44] han presentado métodos que se encargan de predecir la posibilidad de abandono por parte de los estudiantes en una fase temprana del curso para poder tomar las acciones preventivas necesarias. Estos modelos predictivos se entrenan en base a datos que representan la actividad y el progreso de los usuarios, haciendo posible la obtención de predicciones dinámicas y adaptadas al rendimiento y progreso de cada estudiante en ese momento del curso.

Los estudios relacionados se pueden dividir en dos grupos, en función del tipo de los datos que emplean. El primer grupo se caracteriza por emplear datos de características invariantes en el tiempo para determinar las variables más importantes en la predicción del estado de abandono de los estudiantes, como pueden ser las características demográficas y el rendimiento académico anterior. El segundo grupo incorpora atributos variantes en el tiempo, como podría ser el progreso del estudiante.

Muchos estudios basados en datos invariantes utilizan métodos de regresión logística para construir modelos predictivos. En uno de ellos se hizo uso de la combinación de cuatro variables o predictores extraídos de los datos de 2162 estudiantes, y se obtuvo un 79.3% de acierto en la clasificación de usuarios con riesgo de abandono [45]. En dicho caso los predictores invariantes eran edad, disponibilidad de ordenador en casa, nota media reportada y el periodo de trabajo en el curso e-learning.

Otros predictores tenidos en cuenta en otros estudios son el género, etnia, solicitud de beca o ayuda económica, nivel de estudio de los padres o nivel de uso de medios tecnológicos y experiencia previa en el campo de la informática [46]–[49].

Por otro lado, hay estudios que se han basados en variables institucionales que pueden afectar en la retención de los estudiantes, como puede ser el número de alumnos matriculados, gasto económico de la institución en la instrucción y el apoyo académico, el

tamaño de la población dónde está ubicada la institución y la tasa de desempleo en dicha región [50].

Además del uso características del pasado del estudiante o invariantes en el tiempo para predecir la posibilidad de abandono, otros estudios también utilizan características variantes en el tiempo, como se ha mencionado anteriormente. Muchos de estos estudios combinan el uso de datos invariantes y datos correspondientes al progreso del estudiante en ese momento del curso [51]–[55]. Todos ellos hacen uso de técnicas de machine learning o aprendizaje automático como pueden ser las redes bayesianas, árboles de decisión, máquinas de vector soporte o Support Vector Machines (SVM), redes neuronales, etc.

Otro estudio hizo uso de información variante más detallada, la cual reflejaba la actividad diaria del estudiante dentro del curso e-learning [56]. En este caso se utilizaron tres técnicas de aprendizaje automático: FFNN, SVM y probabilistic ensemble simplified fuzzy ARTMAP (PESFAM). La tarea de clasificación consistía en un problema binario, donde la clase 0 representa que el estudiante va a completar el curso y la clase 1 lo contrario. Dado que una sola técnica de aprendizaje automático puede no detectar de manera muy precisa todos los casos de estudiantes propensos a abandonar el curso, mientras otra puede tener éxito en dichos casos concretos, combinar las decisiones de los métodos anteriormente citados bajo un mismo esquema de decisión fue la estrategia de este estudio.

Primero, los resultados predichos (0 o 1) por cada uno de los tres métodos se suman para calcular el nivel de abandono de cada estudiante del conjunto de datos. Por lo tanto, el nivel de abandono de un estudiante puede tener un valor dentro del rango 0-3, donde 0 significa que va a completar el curso y 3 significa que todos los métodos lo han categorizado como caso de abandono.

Una vez calculado el nivel de abandono de un estudiante, este queda categorizado como estudiante finalizador o caso de abandono utilizando tres posibles esquemas de decisión diferentes (Figura 8):

- Esquema 1: se considera que un estudiante va a abandonar el curso si por lo menos uno de los tres métodos así lo clasifica. Es decir, un estudiante es clasificado como caso de abandono si el nivel de abandono calculado es mayor o igual a 1.

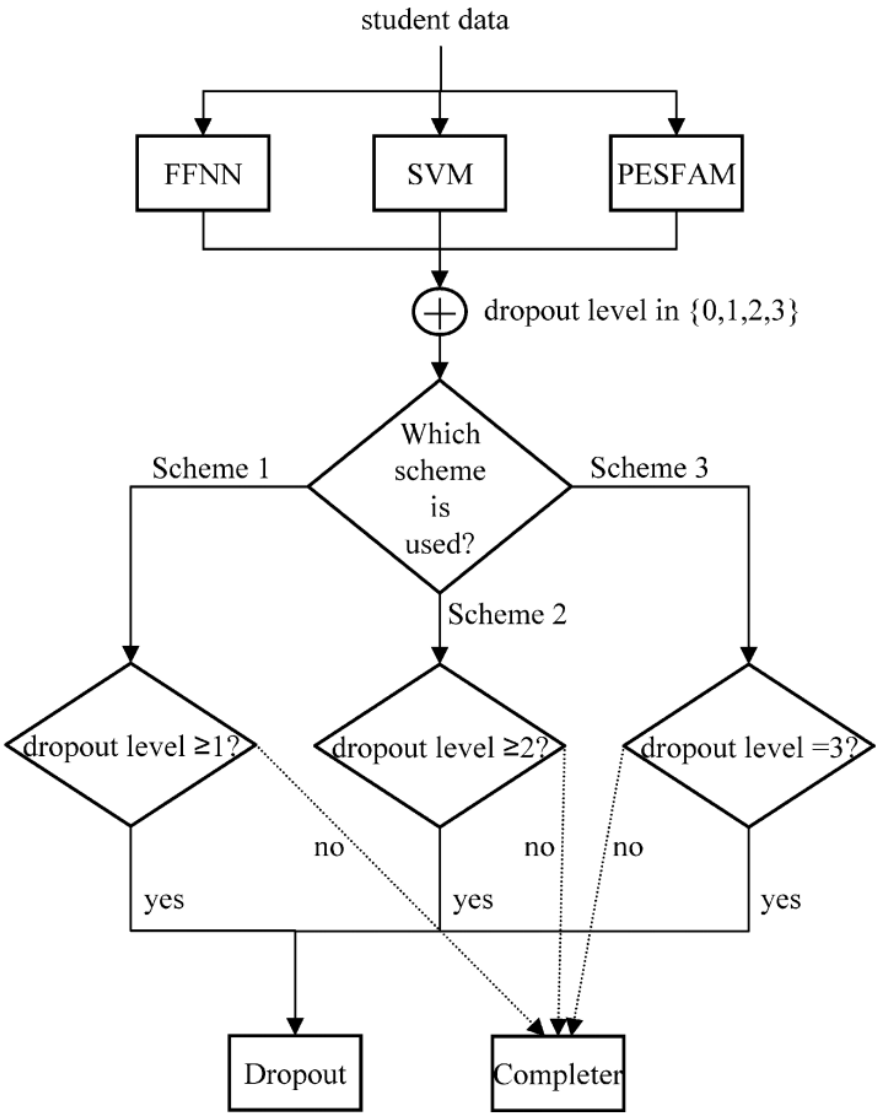

Figura 8. Diagrama de los esquemas de decisión planteados por el estudio
Fuente: [56]

- Esquema 2: se considera que un estudiante va a abandonar el curso si dos de los tres métodos así lo clasifican. Es decir, un estudiante es clasificado como caso de abandono si el nivel de abandono calculado es mayor o igual a 2.
- Esquema 3: se considera que un estudiante va a abandonar el curso si los tres métodos así lo clasifican, Es decir, un estudiante es clasificado como caso de abandono si el nivel de abandono calculado es igual a 3.

Para medir el rendimiento de cada uno de los métodos de aprendizaje automático empleados y de los esquemas de decisión planteados, se llevaron a cabo las predicciones sobre el conjunto de datos de prueba o test y se extrajeron las medidas de accuracy, sensitivity y precision.

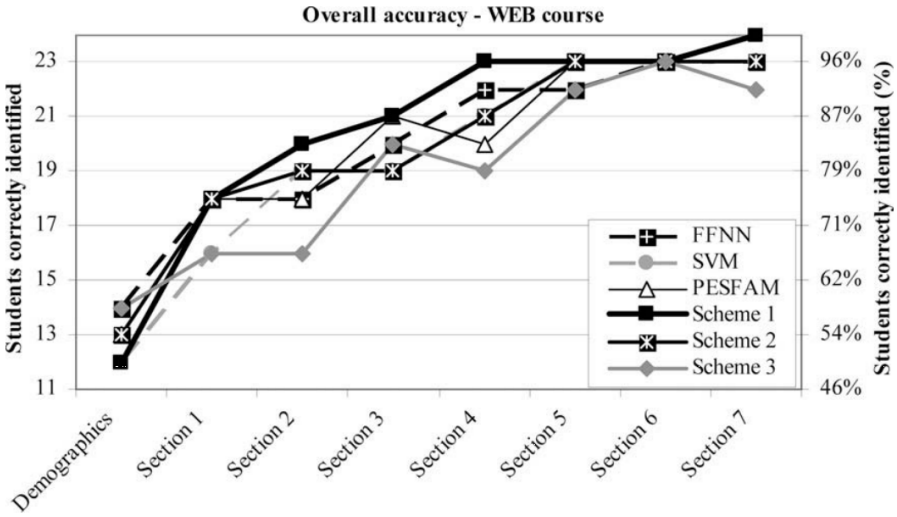

Figura 9. Accuracy obtenida con los diferentes métodos y esquemas de decisión
Fuente: [56]

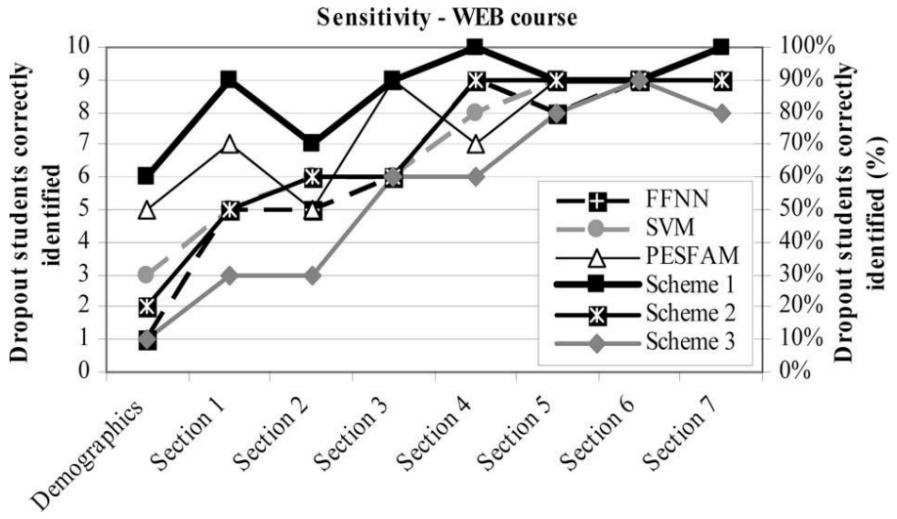

Figura 10. Sensitivity obtenida con los diferentes métodos y esquemas de decisión
Fuente: [56]

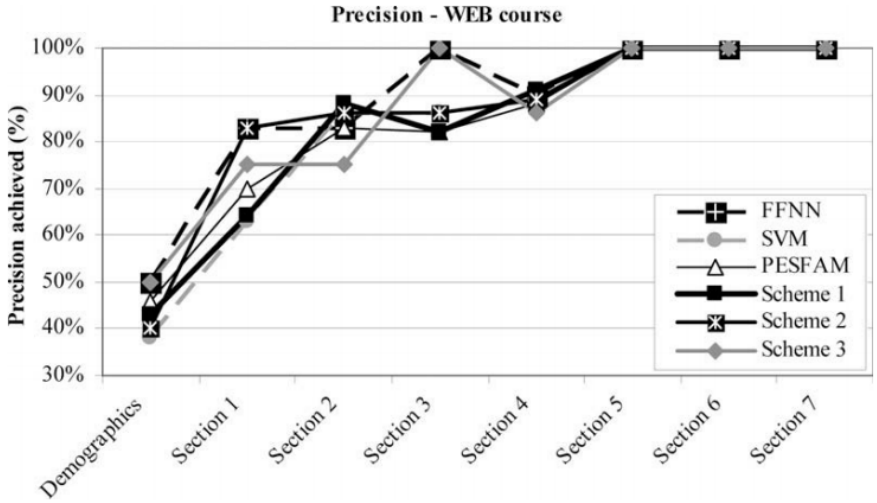

Figura 11. Precision obtenida con los diferentes métodos y esquemas de decisión
Fuente: [56]

La Figura 9 representa gráficamente los resultados de accuracy obtenidos. En dicha gráfica el eje vertical izquierdo representa el número de estudiantes correctamente predichos por cada técnica, mientras que el eje vertical derecho representa el ratio o porcentaje de acierto. Por último, el eje horizontal corresponde a los pasos sobre los cuales el algoritmo ha sido probado, empezando por las características demográficas hasta llegar a la séptima sección del curso. Al principio, como es de esperar, tomando las características demográficas como única información de entrada al algoritmo hace que las diferentes técnicas empleadas no proporcionen un rendimiento mínimamente significativo. Sin embargo, a medida que el curso progresa, los resultados de predicción de todas las técnicas mejoran significativamente.

Los resultados demuestran que ningún método predictivo empleado en solitario obtiene el mejor valor de accuracy en ninguna sección del curso, lo cual significa que el uso de esquemas de decisión ayuda a mejorar el accuracy. Concretamente, el esquema con mejor rendimiento ha sido el 1, logrando alcanzar un mayor accuracy en todas las secciones, empezando con un 75% en la primera y alcanzando el 100% en la última. Por lo tanto, la implementación de las técnicas de aprendizaje automático sobre los datos del e-learning, y especialmente su combinación con el Esquema 1 de decisión, han dado lugar a un alto rendimiento en la predicción del estado de finalización o abandono de los estudiantes, incluso desde las primeras secciones del curso. Sin embargo, no se encontró que las características demográficas por sí solas ayuden a predecir con alta precisión si un estudiante abandonará el curso o no.

Por otro lado, es posible que la tasa de accuracy no refleje completamente las capacidades del método sobre la tarea de predicción de abandono, ya que tiene en cuenta predicciones correctas tanto de casos de finalización del curso como de abandono. Para medir de manera más concisa este aspecto se dispone de los criterios de las métricas de sensitivity y precision.

La gráfica de la Figura 10 refleja los resultados de sensitivity, donde el eje vertical derecho representa el porcentaje de casos de abandono acertados por cada método, el eje vertical izquierdo el número de casos de abandono predichos correctamente, y el eje horizontal representa los pasos sobre los cuales el algoritmo ha sido probado, empezando por las características demográficas hasta llegar a la séptima sección del curso.

En dicha gráfica se puede observar que los resultados durante las dos primeras secciones del curso son por lo general bajos, siendo otra vez el Esquema 1 de decisión el método con mejores resultados. Este hecho vuelve a remarcar que, a pesar de los fallos de los métodos individuales, es difícil que la clasificación de un caso de abandono pueda ser fallada por los tres métodos al mismo tiempo, lo cual hace que los esquemas de decisión jueguen un papel fundamental en este estudio. Por otro lado, a medida que se avanza en las secciones del curso más información se recoge acerca de los estudiantes, y por lo general todos los métodos muestran una mejora en sus resultados.

Por último, la Figura 11 refleja los resultados de precision donde el eje vertical corresponde al porcentaje de precision alcanzado por cada método y el eje horizontal los pasos sobre los cuales el algoritmo ha sido probado. En primer lugar, como se puede observar, los resultados de precision obtenidos con las características demográficas son muy bajos, al igual que ocurría en el caso de las dos métricas anteriores. Por otro lado, en este caso el Esquema 1 de decisión refleja resultados más bajos en las secciones 1 y 3 respecto a los casos anteriores. Sin embargo, en las secciones 5, 6 y 7 muestra unos resultados iguales o superiores a dichos casos.

Se ha podido comprobar que este esquema de decisión clasificó erróneamente a varios estudiantes que completaron el curso, pero identificó correctamente un número significativamente mayor de casos de abandono. Por ejemplo, en la sección 3 este esquema obtuvo un 82% de precision, proveniente de clasificar correctamente nueve casos de abandono e incorrectamente dos casos que completaron el curso. Para la misma sección la técnica FFNN ha clasificado correctamente todos los estudiantes que han completado el curso, obteniendo un 100% de precision, pero solo identificó 6 casos de abandono. Por lo tanto, a pesar de proporcionar resultados de menor precision que la técnica FFNN, el Esquema 1 de decisión es preferible para los propósitos de este estudio donde es más importante detectar casos correctos de abandono.

Otro elemento importante en la predicción de abandono es la pronta clasificación, es decir, la capacidad de identificar a un estudiante con riesgo de abandono antes de que realmente este decida abandonar el curso. Una predicción de abandono en una fase oportuna permite facilitar a los instructores una pronta intervención, proporcionando ayuda a los estudiantes

y permitiendo que completen sus estudios. Por este motivo, para cada alumno que abandonó el curso, se calculó cuál fue el último tramo del curso en el que este mostró progreso o actividad y la sección calculada se comparó con la sección en la que el método propuesto identificó correctamente al estudiante como caso de abandono.

Los resultados obtenidos por el estudio muestran que la sección promedio en la que mostraron progreso por última vez los alumnos que abandonaron el curso fue la 4.6 (de las 7 secciones que componen la estructura del curso e-learning), lo cual demuestra que los alumnos en promedio participan en varias secciones antes de abandonar los cursos. La sección promedio en la que el método propuesto descubre en primer lugar el posible caso de abandono es la 1.5. Esto significa que el método propuesto por el estudio identifica a los estudiantes propensos al abandono en un promedio de 3.1 secciones antes de la última sección en la que estos estudiantes muestran actividad/interés y, por lo tanto, se espera que ayude a los instructores a intervenir rápidamente para ayudar a estos estudiantes.

5. Objetivos

El objetivo principal de este trabajo es desarrollar un sistema predictivo a partir del cual, mediante el progreso del estudiante en el curso e-learning, se pueda predecir cuál va a ser la nota que va a obtener dicho estudiante en la próxima prueba de evaluación o test. Gracias a esto los estudiantes podrán conocer con antelación su nivel de preparación de cara al próximo examen y, por otro lado, los docentes o instructores podrán detectar en una etapa temprana quiénes son los alumnos con menor preparación de cara a la próxima prueba. Esto permitiría proporcionar una asistencia de apoyo personalizada a los estudiantes que se encuentren en dicha situación, evitando así los efectos de cualquier barrera de aprendizaje que pueda provocar una incompleta formación o incluso un abandono del curso.

Para lograr este objetivo principal, se plantean los siguientes subobjetivos:

- Disponer de un conjunto de datos adecuado: en primer lugar, se tendrá que llevar a cabo un proceso de tratamiento de los datos provenientes de la base de datos del curso e-learning del cual partimos. Este proceso va a conllevar diferentes fases de limpieza, depuración y agregación, dando como resultado final un conjunto de datos preparado para ser utilizado en el entrenamiento del modelo o modelos predictivos.

  Tras el tratamiento de los datos se medirá si la cantidad obtenida es suficiente para llevar a cabo un proceso de entrenamiento adecuado. En caso de no ser así, el siguiente paso será la generación de datos sintéticos a partir de las características del entorno de los datos reales para complementar a estos últimos. De esta manera se podrá alcanzar una cantidad de datos suficiente para continuar con el objetivo principal de este trabajo.
- Modelar el sistema y determinar la mejor solución: tras obtener los datos el siguiente paso es la implementación del modelo predictivo. Para ello habrá que escoger cuáles son las técnicas o métodos de aprendizaje automático más adecuados para dar solución a nuestro problema.

Una vez escogidas e implementadas las principales técnicas de aprendizaje automático que permiten abordar la solución, el último paso es la comparación de los resultados o rendimiento obtenido con cada una de ellas, lo cual permitirá escoger la solución final más adecuada.

6. Metodología

Para abordar el trabajo de este proyecto se va a seguir una metodología secuencial, en la cual se pasa de una fase a otra una vez completada la inmediatamente anterior y donde hay un orden preestablecido para la realización de cada una de ellas. En la Figura 12 se representan las fases que componen esta metodología y el orden en el que se realizan.

En primer lugar, se comienza con un estudio teórico que nos permita adquirir todos los conocimientos necesarios acerca de los factores principales implicados en la solución de nuestro problema. Este estudio se ha reflejado en el apartado 4 de este trabajo (estado de la cuestión), donde se plantean los antecedentes y el punto de partida de este trabajo junto al estudio de las diferentes soluciones actualmente disponibles con sus metodologías y técnicas empleadas para abordar problemas similares al nuestro.

La segunda fase conlleva el diseño y automatización del flujo de preprocesamiento de los datos provenientes del curso e-learning monitorizado en el proyecto antecedente. Esta fase es de vital importancia, debido a que conlleva el estudio y selección de las variables que se consideran más importantes para tener en cuenta en la implementación del modelo predictivo. En tercer lugar, se encuentra la fase de generación de datos, la cual es opcional en función del tamaño del conjunto final obtenido en la fase anterior.

Una vez completadas las fases relacionadas con la manipulación de los datos, tanto de preprocesamiento como de simulación, se avanza a la fase de implementación en la cual se engloban todos los procedimientos necesarios para el desarrollo de los diferentes modelos predictivos, en función de los diferentes métodos y técnicas disponibles para solucionar nuestro problema. Finalmente, se lleva a cabo la fase de comparativa y conclusión, en la cual se compara el rendimiento de los diferentes modelos y técnicas aplicadas, para finalmente seleccionar el modelo predictivo final de nuestra solución.

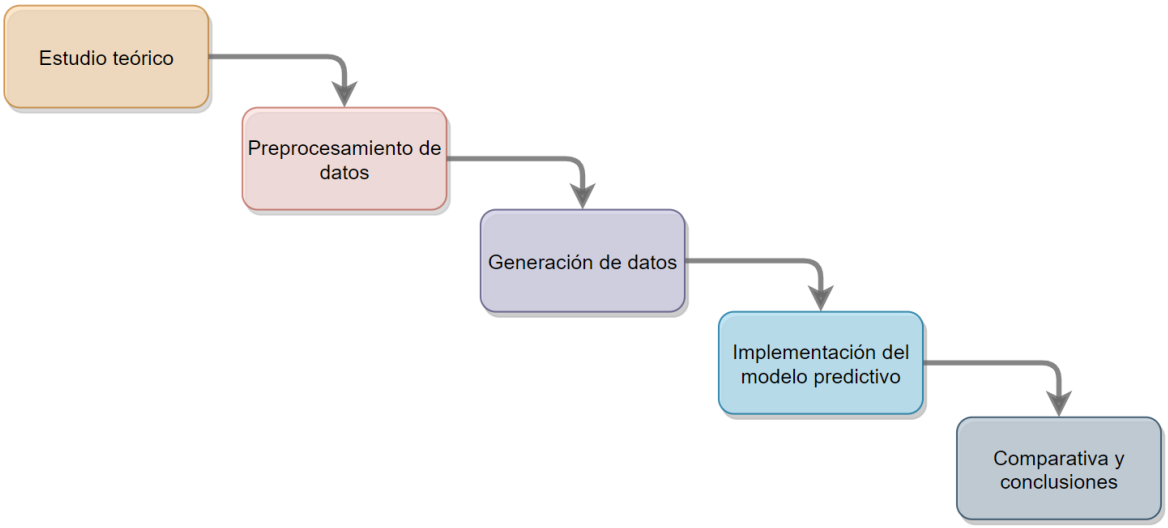

Figura 12. Metodología seguida en el trabajo

7. Análisis y tratamiento de los datos

En este apartado se va a llevar a cabo todo el proceso que engloba el tratamiento de los datos a partir de los cuales se implementará posteriormente el sistema predictivo. Concretamente se trabajará en la construcción de la estructura de datos final, con su correspondiente grupo de variables explicativas y la variable a predecir.

7.1. Preprocesamiento de los datos

Como se ha explicado en el apartado de antecedentes, nuestro punto de partida van a ser los datos monitorizados en un curso e-learning sobre el consumismo y cambio climático. De dicho curso se hicieron dos promociones, donde se consiguió el registro de 24 y 3 usuarios respectivamente. En este curso se podían monitorizar diversos eventos relacionados con diferentes actividades del usuario: *ha empezado, ha avanzado a, ha vuelto a, ha navegado a, ha observado, ha revisado* y *ha contestado.* Como se ha ido explicando durante los diferentes apartados introductorios y en el planteamiento de los objetivos, la finalidad de este proyecto es crear un modelo capaz de predecir, en función del progreso registrado mediante los eventos anteriormente mencionadas, el rendimiento del usuario en la próxima prueba de evaluación o test.

El curso tiene un total de 3 tests, y para cada uno de ellos se implica el conocimiento de un número de lecciones diferentes. A partir de los diferentes tipos de eventos registrados, para cada test, vamos a calcular las siguientes variables de actividad:

- Número de revisiones realizadas en cada lección. Esta variable se puede medir a través del evento *ha revisado*, el cual registra para cada usuario la lección revisada y el momento en el que lo ha realizado. Por lo tanto, el objetivo va a ser contabilizar todas las revisiones realizadas por el usuario a cada lección implicada en el test en cuestión.
- Tiempo total dedicado a las lecciones de cada test. Para calcular este tiempo se necesita saber la hora en la cual el usuario ha observado por primera vez la primera

lección implicada en el test y, por otro lado, la hora en la que finaliza el test en cuestión. Con estas dos horas, y teniendo en cuenta que las lecciones se desbloquean una vez completada la inmediatamente anterior, se puede extraer el periodo de tiempo dedicado a las lecciones implicadas en el test. Dichas horas se pueden obtener a partir de los eventos *ha observado* y *ha respondido*, donde el primero nos indica la primera vez que el usuario observa una lección en concreto y el segundo el momento en el cual el usuario responde a la cuestión de un test, escogiendo en este caso el momento de la respuesta a la última pregunta del test.

- Número de navegaciones realizadas. Esta variable se puede medir a través del evento *ha navegado a*, el cual registra las navegaciones del usuario a cualquier lección. Por lo tanto, el objetivo va a ser contabilizar todas las navegaciones realizadas por el usuario a cualquiera de las lecciones implicadas en el test en cuestión.
- Número de aciertos obtenidos en el test. A través del evento *ha contestado* se registra la respuesta del usuario a las preguntas de un test, indicando si ha sido correcta o incorrecta. Por lo tanto, el objetivo va a ser contabilizar el número de respuestas correctas que obtiene el usuario en el test en cuestión.

Con las variables anteriores se puede obtener información muy detalla acerca del progreso del usuario antes de llegar a cada test. Estas variables serán las empleadas en el proceso de entrenamiento del modelo de aprendizaje automático, mediante el cual se predecirá la variable que representa el número de aciertos. No obstante, todas ellas requieren de un proceso de síntesis o cálculo a través de los diferentes eventos registrados. Por ello, en este apartado se va a llevar a cabo el preprocesamiento de los datos mediante la herramienta Pentaho Data Integration[5], la cual permite llevar a cabo diversas transformaciones de limpieza, depuración y estructuración de datos. Esta tarea se va a organizar en flujos de preproceso independientes llamados transformations o transformaciones, donde cada una lleva a cabo la extracción y estructuración de cada una las variables mencionadas. Por último, se implementará un flujo global, llamado job o trabajo que invoca de manera

[5] https://help.hitachivantara.com/Documentation/Pentaho/7.1/0D0/Pentaho_Data_Integration

secuencial y en orden de dependencia a todas las transformaciones independientes, dando como resultado final el conjunto de datos esperado.

7.1.1. Aciertos

Para la extracción de la variable "aciertos" se implican dos flujos o transformaciones. En primer lugar, se lleva a cabo la transformación de la Figura 13, en la cual el flujo comienza con la lectura de los dos archivos que contienen los registros del evento *ha respondido* de ambas promociones del curso en formato CSV. Una vez leídos estos archivos, se concatena la información de ambos bajo una misma estructura.

Debido a la gran cantidad de metadatos contenidos en estos registros, el siguiente paso es seleccionar únicamente las columnas de interés: identificador del usuario, timestamp o marca temporal, identificador del test, número de la pregunta respondida y resultado. Tras seleccionar las columnas de interés es necesario ajustar los formatos de ciertos campos a nuestras necesidades, como puede ocurrir con el identificador del test. Este identificador se almacena como "undefined – Test x" donde *x* es el número del test (1, 2 o 3).

De dicho campo solo nos interesa conocer la subcadena "Test x", por lo que se aplica una acción de tipo split o división donde la cadena se divide en las dos partes separadas por el guión, quedándonos finalmente con la segunda parte. Lo mismo ocurre con el campo timestamp, el cual contiene en una misma cadena la fecha y la hora bajo el formato "aaaa-MM-ddT1hh:mm:ssZ". En este caso nos interesa tener la fecha y la hora separadas en columnas independientes, por lo que se aplica también una acción de split dividiendo la cadena en las dos partes separadas por el indicador "T1". Una vez separadas, falta eliminar el carácter "Z" situado al final de la cadena de la hora. Para ello se hace uso de una acción cut o corte, la cual permite cortar parte de la cadena según la posición indicada.

El último ajuste aplicado en este flujo es el correspondiente al número de la pregunta del test, el cual viene acompañado con el enunciado de esta en el siguiente formato: "nº. Enunciado". En este caso, haciendo uso de nuevo de la acción cut, se extrae de esta cadena solo el identificador de la pregunta sin su enunciado.

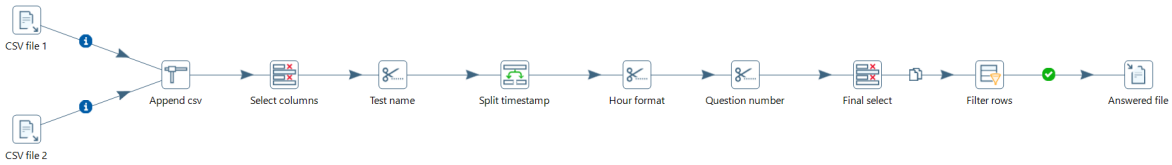

Figura 13. Flujo de transformación para la extracción de la variable aciertos (1)

Finalmente, se hace una selección de columnas para descartar aquellas resultantes de las acciones de split, seguido de un filtrado de filas donde se seleccionan solo aquellas no generadas por el usuario administrador. Con esto último el objetivo es evitar datos registrados a causa de acciones de administración o prueba, sin pertenecer realmente al progreso de un alumno real. Con todas estas acciones queda depurada la estructura de registros del evento *ha respondido*, la cual en el paso final se almacena en un archivo CSV.

La Figura 14 refleja el segundo flujo correspondiente a la extracción de la variable de "aciertos". Este flujo parte de la estructura de datos almacenada anteriormente, la cual una vez depurada va a ser útil para agrupar todas las instancias generadas por cada uno de los usuarios en función del test realizado, extrayendo así el número de aciertos totales obtenidos en cada prueba. Por lo tanto, el primer paso del flujo es la lectura del archivo CSV almacenado por la primera transformación.

Para llevar a cabo la acción de agrupamiento, la herramienta Pentaho Data Integration exige ordenar las instancias en función de los campos que serán empleados para dicha agrupación. Por ello, se hace uso de una acción sort u ordenación, en la cual se ordenan todas las instancias primero en función del identificador del usuario y después por el identificador del test. Seguidamente se emplea una acción group by o agrupación, donde los campos de agrupación son justamente los anteriormente mencionados.

Dentro del proceso de agrupación se ha indicado la generación de las siguientes agregaciones o variables: suma de los valores del campo de aciertos (tiene valores 0 o 1, en función de si ha sido fallo o acierto), conteo de las filas correspondientes a las preguntas respondidas y extracción del valor máximo del campo de la hora. Con esto se tendrá para cada usuario el número de aciertos obtenidos, el número total de preguntas respondidas y la hora en la que ha respondido a la última pregunta de cada test.

Una vez realizada la agrupación y la generación de las correspondientes agregaciones, se filtran las instancias correspondientes a cada test y se almacenan en un archivo CSV

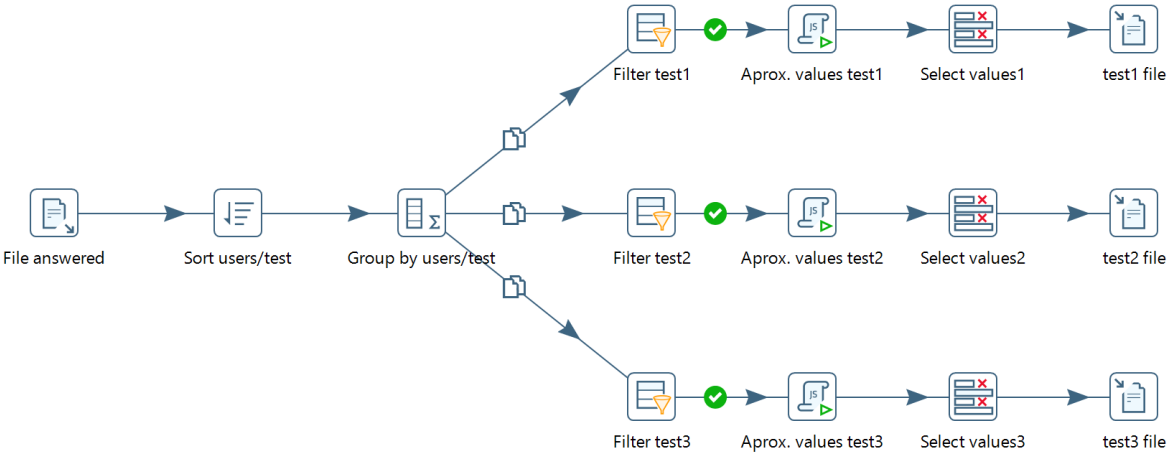

Figura 14. Flujo de transformación para la extracción de la variable aciertos (2)

independiente. Es decir, el resultado de este flujo son tres archivos CSV con los aciertos de cada usuario en cada uno de los tres tests.

Para poder tratar de manera igualitaria a los valores de la variable "aciertos" habría que almacenar la agregación anterior solo para aquellos usuarios que han completado el test, es decir, para aquellos que han respondido a todas las preguntas. Sin embargo, para no reducir de manera excesiva el tamaño de nuestro conjunto de datos descartando los casos de tests incompletos, en esta solución se ha decidido extrapolar estos valores. La técnica empleada ha sido la regla de tres, de manera que si el test tiene un total de seis cuestiones y un usuario solo ha respondido a cuatro de ellas obteniendo tres aciertos, se extrapola este valor al equivalente habiendo respondido las seis cuestiones. Para llevar a cabo esta operación se ha hecho uso de la ejecución de un pequeño script en lenguaje Javascript dentro de las acciones permitidas por Pentaho Data Integration.

7.1.2. Revisiones

El flujo de la Figura 15 representa el proceso de transformación llevado a cabo para extraer los valores de la variable "revisiones". En esta segunda transformación, y en las próximas, el objetivo va a ser extraer el valor de la variable en cuestión y añadirla a la estructura de datos obtenida en el apartado anterior. Por lo tanto, se comienza con la lectura de los dos archivos CSV que contienen los registros del evento *ha revisado*, se concatena el contenido

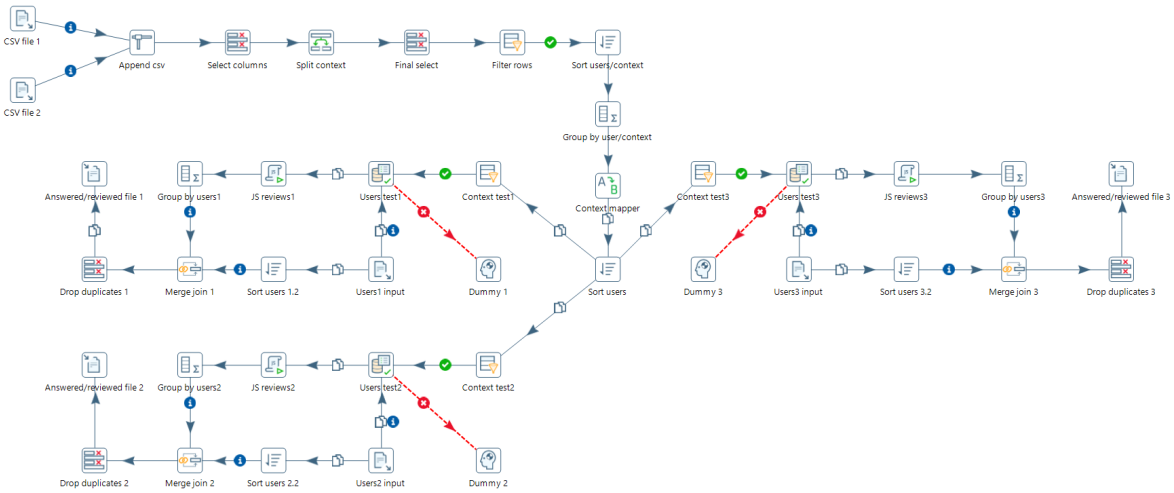

Figura 15. Flujo de transformación para la extracción de la variable revisiones

de ambos archivos en una misma estructura y se seleccionan las columnas de interés: el identificador del usuario y el nombre de la lección revisada.

El nombre de la lección revisada viene en formato "undefined - Nombre", donde solo nos interesa la subcadena "Nombre". Por ello, se hace uso de una acción split, a partir de la cual se divide la cadena en las dos partes separadas por el guión, quedándonos finalmente con la segunda subcadena. Una vez formateada esta columna y filtradas las filas generadas por usuarios no administradores, se procede con la agrupación de los datos en base al identificador del usuario y el nombre de la lección. Con esto, el objetivo es generar una nueva columna agregada como resultado del conteo de las filas en las cuales aparece el mismo nombre de lección por usuario, consiguiendo así el número de revisiones realizadas por cada usuario en cada lección.

Para mejorar la legibilidad del nombre de las lecciones se hace uso de una acción de tipo value mapper o mapeador de valores, en la cual se mapean todos los nombres de las lecciones por sus correspondientes identificadores abreviados. Estos identificadores tienen el formato "sxxlyy", donde *sxx* indica el número de la sección (bloque) y *lyy* indica el número de la lección dentro de dicho bloque. Por ejemplo, s02l01 significa lección 1 de la sección 2.

A partir de este punto la estructura de datos contiene tres columnas: usuario, lección y revisiones, y el siguiente paso sería añadir esta nueva información a la estructura de datos obtenida en el apartado anterior. Recordemos que en dicha transformación los resultados se almacenaron en tres archivos diferentes, en función del test al que pertenecen. Por ello,

el primer paso es filtrar las filas de la estructura actual en función del test al que pertenece la lección revisada.

Una vez filtrados los datos por tipo de test, es necesario validar que las revisiones pertenezcan a usuarios participantes en dichos tests. Es decir, puede haber registros de revisiones de usuarios que no han llegado a realizar ningún test en concreto, lo cual es información que no nos interesa mantener. Para llevar a cabo esta validación se hace uso de la acción data validator o validador de datos, la cual recibe dos estructuras de datos; la primera estructura es la actual y la segunda es la correspondiente al apartado anterior. De esta manera, la columna de usuarios de la segunda estructura se emplea para validar la columna de usuarios de la estructura actual, manteniendo así únicamente las revisiones de usuarios que han realizado el test en cuestión.

Tras validar los datos, se ha decidido modificar la jerarquía de la estructura actual mediante la ejecución de un script en lenguaje Javascript. Con esta modificación, en lugar de tener las columnas "lección" y "revisiones", el identificador de cada lección pasa a ser una columna cuyos valores son las revisiones realizadas por cada usuario en dicha lección. Gracias a esta nueva estructura disponemos de las revisiones de cada lección como una columna o variable independiente, lo cual es necesario de cara a la futura implementación del modelo predictivo.

Por último, se une la estructura actual junto a la obtenida en el apartado anterior, y para ello se hace uso de la acción merge join o unión. A través de esta acción se han podido unir las columnas de ambas estructuras empleando la columna del identificador del usuario como punto de enlace entre ellas. Para finalizar el flujo, se almacena la estructura correspondiente a cada test en un archivo independiente, donde ahora se dispone de las siguientes columnas: identificador del usuario, aciertos, revisiones de cada lección implicada en el test y la hora de finalización del test.

7.1.3. Navegaciones

Para la extracción de esta variable se ha implementado el flujo de la Figura 16. En primer lugar, se leen los archivos que contienen los registros del evento *ha navegado a* y se concatena la información de ambos bajo una misma estructura. Seguidamente se seleccionan las columnas de interés, siendo en este caso el identificador del usuario y el nombre de la lección a la cual se navega.

Al igual que ocurría en la transformación del apartado anterior, para hacer más legible el nombre de las lecciones, se hace un mapeo de los nombres de estas por sus correspondientes identificadores abreviados. El siguiente paso es filtrar los datos en función de las lecciones que pertenecen a cada test en concreto y, una vez separados, se hace una agrupación de los datos en base al identificador de usuario generando como agregación el número de filas agrupadas. Con esto se obtiene para cada test una estructura formada por la columna del identificador del usuario y otra columna con el número de navegaciones totales realizadas a las lecciones implicadas en dicho test.

Una vez obtenida esta estructura es necesario añadir su información a la estructura obtenida en el apartado anterior, la cual hasta ahora contiene las variables de aciertos y revisiones. Para ello, primero se validan los datos de navegación para usuarios que hayan participado en el test en cuestión y posteriormente se hace la unión de ambas estructuras, haciendo uso de las mismas acciones empleadas en el apartado anterior.

Por último, se almacena la estructura correspondiente a cada test en un archivo CSV independiente, donde ahora se dispone de las siguientes columnas: identificador del usuario, aciertos, revisiones de cada lección implicada en el test, número de navegaciones realizadas y la hora de finalización del test.

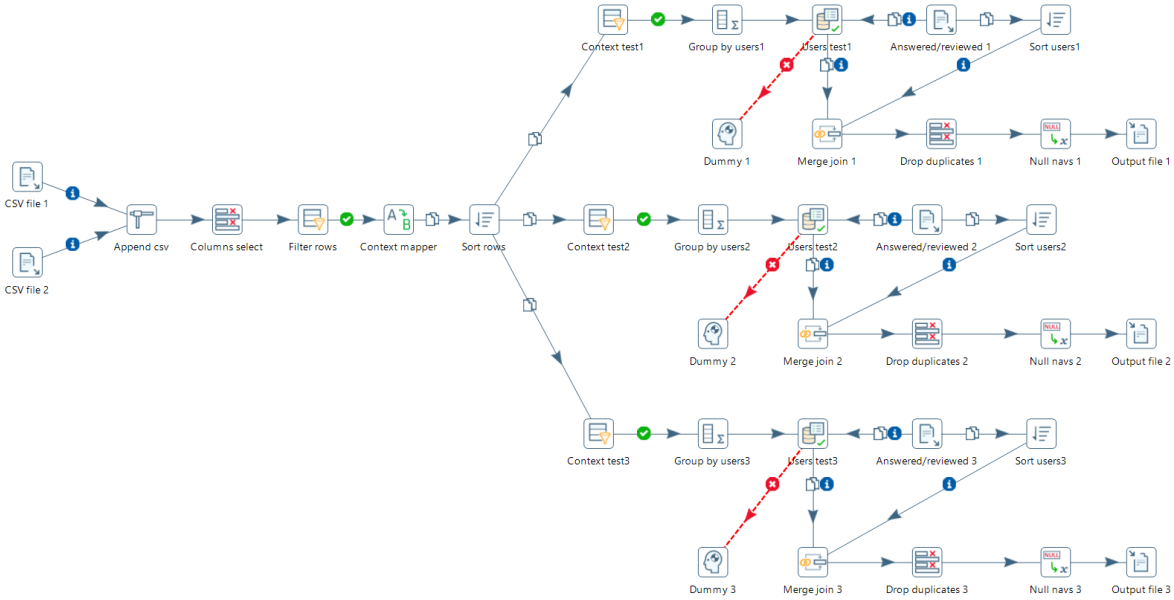

Figura 16. Flujo de transformación para la extracción de la variable navegaciones

7.1.4. Tiempo

En la Figura 17 se muestra el flujo de la transformación empleada para la extracción de la variable "tiempo". Para calcular el tiempo invertido en cada test y sus lecciones correspondientes se siguen dos procedimientos diferentes.

Para el primer test se hace uso de los registros almacenados a raíz del evento *ha observado*, por lo que en primer lugar se leen los dos archivos CSV correspondientes y se concatenan bajo un misma estructura. De esta estructura de datos se seleccionan tres columnas: identificador del usuario, lección observada y la hora en la que se genera el evento. Con esto, el objetivo es filtrar posteriormente la hora en la que cada usuario observa por primera vez la primera lección implicada en el Test 1. Para ello, primero se debe formatear la columna del nombre de la lección, la cual tiene el formato "undefined – Nombre de la lección". Por lo tanto, mediante una acción split se divide la cadena en las dos partes separadas por el guión y nos quedamos con la segunda. Con la columna de la hora o timestamp ocurre lo mismo (formato "aaaa-MM-ddT1hh:mm:ssZ"), se hace uso de la acción split para dividir la cadena en las dos partes separadas por el identificador "T1", nos quedamos con la segunda subcadena y de esta se corta la parte correspondiente a la hora.

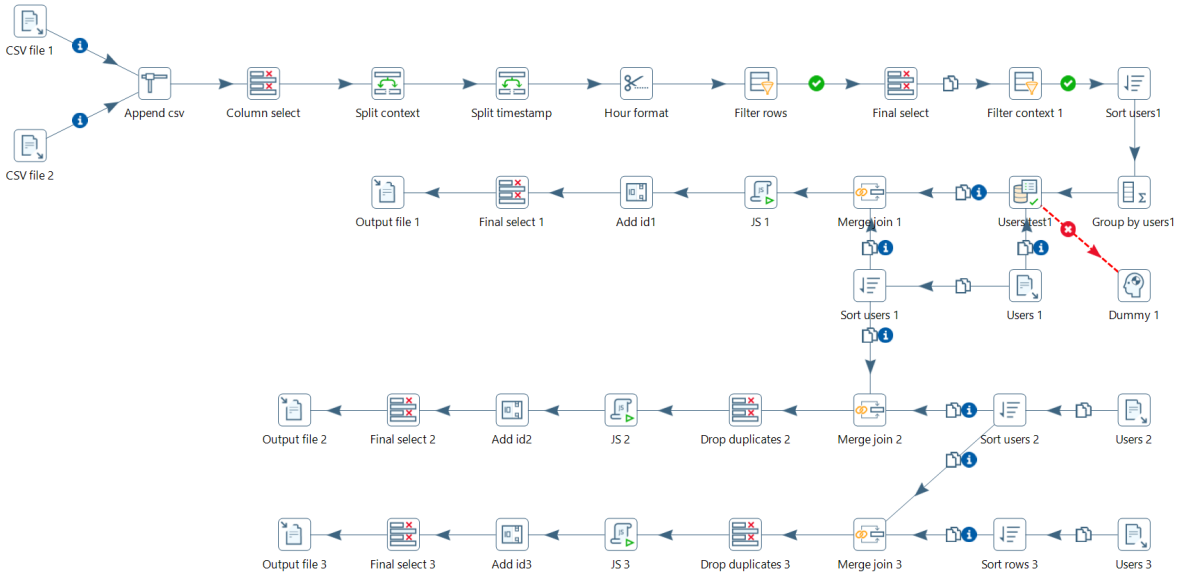

Figura 17. Flujo de transformación para la extracción de la variable tiempo

Figura 18. Flujo final para la automatización del preprocesamiento de los datos

El siguiente paso, es validar los datos correspondientes a usuarios participantes en el Test 1, como se ha hecho en apartados anteriores, cargando la estructura de datos obtenida para este test en el apartado anterior. Además, con la carga de esta segunda estructura se puede acceder a la columna que representa la hora en la cual el usuario ha finalizado el Test 1.

Por lo tanto, teniendo la hora en la que se observa por primera vez la primera lección implicada en el Test 1 y la hora en la que se finaliza este, se puede extraer mediante la ejecución de un script en lenguaje Javascript el tiempo, en minutos, dedicado a la preparación del Test 1. Como resultado final, se añade esta nueva columna a la estructura de datos correspondiente al Test 1 del apartado anterior y se almacena en un archivo CSV con las siguientes columnas: identificador del usuario, revisiones por lección, navegaciones, tiempo invertido y aciertos.

Para la extracción del tiempo invertido en los tests 2 y 3 se sigue el mismo procedimiento anterior, pero con una única diferencia. En este caso en lugar de hacer uso de los registros del evento *ha observado*, directamente se toma como hora de inicio la hora de finalización del test anterior. Es decir, para calcular el tiempo invertido en el Test 2 se calcula el periodo de tiempo transcurrido entre la hora en la que se finaliza el Test 1 y la hora en la que se

finaliza el Test 2, y lo mismo aplicaría para el Test 3 tomando la hora de finalización del Test 2 como hora de inicio.

7.1.5. Proceso global

Con los apartados anteriores se han podido extraer todas las variables necesarias que representan el progreso de los usuarios a partir de los datos almacenados por la generación de los diferentes eventos de actividad disponibles en el curso e-learning.

Para finalizar el preprocesamiento de los datos, queda pendiente automatizar todo el proceso mediante un job o trabajo de Pentaho Data Integration. Es decir, el objetivo va a ser emplear un flujo encargado de invocar en orden de dependencia a todas las transformaciones anteriormente citadas y explicadas, obteniendo así la estructura de datos final. Como se puede ver en la Figura 18, el flujo invoca a cada una de las transformaciones en el siguiente orden:

1. Flujo 1 correspondiente a la extracción de la variable "aciertos".
2. Flujo 2 correspondiente a la extracción de la variable "aciertos".
3. Flujo correspondiente a la extracción de la variable "revisiones".
4. Flujo correspondiente a la extracción de la variable "navegaciones".
5. Flujo correspondiente a la extracción de la variable "tiempo".

Por último, se hace uso de una acción encargada de eliminar archivos auxiliares empleados durante los diferentes procesos intermedios. En las tres siguientes tablas (Tabla 2, Tabla 3 y Tabla 4) se pueden observar tres muestras de los datos finales correspondientes a cada uno de los tres tests respectivamente. Como se puede observar, en cada test se implican lecciones diferentes haciendo variar para cada estructura de datos el número de columnas correspondientes a las revisiones.

Tabla 2. Muestra de la estructura de datos final del Test 1

user_id	s02l01	s02l02	navegaciones	tiempo	aciertos
1	9	16	3	3	4
2	3	11	2	2	3
3	7	1	0	2	5
4	2	1	0	7	5
5	1	3	0	1	3

Tabla 3. Muestra de la estructura de datos final del Test 2

user_id	s02l04	s02l05	s02l06	navegaciones	tiempo	aciertos
1	5	2	1	0	2	4
2	0	2	1	0	3	5
3	0	1	5	0	2	5
4	2	1	5	1	4	5
5	4	3	3	0	1	4

Tabla 4. Muestra de la estructura de datos final del Test 3

user_id	s02l08	s02l09	s03l01	s03l02	s03l03	s03l04	s03l05	s03l06	navegaciones	tiempo	aciertos
1	4	2	6	3	1	8	9	2	0	6	5
2	8	5	40	21	2	8	17	2	6	4	4
3	2	0	2	0	1	1	0	0	0	3	3
4	2	2	6	4	3	5	1	2	0	6	5
5	4	5	5	3	5	2	3	3	0	3	3

7.2. Generación de datos sintéticos

Tras el tratamiento de los datos se han obtenido como resultado final tres conjuntos de datos correspondientes a cada uno de los tres tests que componen el curso e-learning. Para el Test 1 se ha obtenido un conjunto de datos con un total de 27 instancias, para el Test 2 se han obtenido 25 instancias y para el Test 3 se han obtenido 24 instancias. Teniendo en cuenta que los métodos de aprendizaje automático requieren de una cantidad de datos bastante

grande para llevar a cabo un correcto proceso de entrenamiento, además de la necesidad de reservar parte de los datos para el proceso de validación o prueba, trabajar sobre un conjunto de datos con menos de 30 instancias se queda muy lejos del mínimo requerido para implementar un modelo predictivo con unos resultados aceptables.

Ante este problema de escasez de datos, siguiendo el plan de contingencia planteado para este riesgo en el apartado de análisis de riesgos y con el objetivo de poder continuar con el desarrollo de este proyecto, se ha decidido emplear técnicas de generación o simulación de datos sintéticos. Concretamente se va a hacer uso de ecuaciones lineales para representar la relación entre las diferentes variables implicadas en la predicción de la variable respuesta, consiguiendo así un escenario ideal para la generación proporcional de los valores de cada una las variables. Para cada test se generará un conjunto de datos con un total de 2000 instancias, lo cual es una cantidad de datos suficiente para implementar de manera adecuada los modelos predictivos correspondientes a cada uno de los tests.

La implementación del proceso de generación de estos datos será programada en lenguaje R[6], debido a que también será empleado para implementar la primera parte del desarrollo de los modelos predictivos, gracias a la facilidad que ofrece para el manejo de estructuras de datos y a su gran variedad de técnicas estadísticas y gráficas.

7.2.1. Datos del Test 1

Las variables que representan el progreso necesario para la predicción del rendimiento en el Test 1 son las siguientes:

- Variables explicativas:
    - s02l01: número de revisiones en la primera lección implicada en el test. Esta variable de aquí en adelante será representada como x_1.
    - s02l02: número de revisiones en la segunda lección implicada en el test. Esta variable de aquí en adelante será representada como x_2.

[6] https://www.r-project.org/

    - tiempo: número de minutos invertidos en las lecciones y en la realización del test. Esta variable de aquí en adelante será representada como x_3.
    - navegaciones: número de navegaciones a cualquiera de las lecciones implicadas en el test. Esta variable de aquí en adelante será representada como x_4.
- Variable respuesta:
    - aciertos: número de cuestiones acertadas de las seis planteadas en el test. Esta variable de aquí en adelante será representada como y.

Una vez conocidas las variables, se puede representar la relación entre todas ellas de la siguiente manera:

$$y = a \cdot x_1 + b \cdot x_2 + c \cdot x_3 + d \cdot x_4 \qquad (1)$$

El objetivo de la ecuación 1 es representar la aportación de cada una de las variables explicativas (x_n) en el resultado final de la variable respuesta (y). La repercusión o peso de cada variable en el resultado final se medirá en función del valor de su correspondiente coeficiente ($a, b, c\ y\ d$) en relación con su escala numérica y, como es lógico, algunas variables tendrán más peso que otras. Estos pesos o coeficientes van a tener valores fijos y, como autores del curso e-learning, vamos a establecer dichos valores en función de la importancia de cada variable en la nota final del test. Para determinar estos valores, el primer paso es establecer el escenario ideal necesario para obtener la máxima nota en el test (6 aciertos). Conociendo el contenido del curso e-learning y observando la media de los valores obtenidos en las variables implicadas por usuarios reales, el escenario ideal para el Test 1 sería el siguiente: un número medio de 10 revisiones por lección ($x_1\ y\ x_2$), 10 minutos como tiempo medio invertido (x_3) y una media de 2 navegaciones (x_4). Con estos valores la ecuación quedaría de la siguiente manera:

$$6 = a \cdot 10 + b \cdot 10 + c \cdot 10 + d \cdot 2 \qquad (2)$$

Una vez establecido el escenario ideal, el siguiente paso es determinar los valores de los coeficientes basándonos en la importancia o repercusión de cada una de las variables en el resultado final. Empezando por la variable de tiempo (x_3), se podría considerar que esta es la más importante de las cuatro variables implicadas en este test, puesto que si no se dedica el tiempo necesario en la lectura y comprensión de las lecciones el resultado del test será

muy deficiente independientemente del resto de variables. Por otro lado, el tiempo dedicado repercute de cierta manera en el valor de las otras variables; cuanto más tiempo invertido mayor número de revisiones se generan, por ejemplo. Por lo tanto, dedicando el tiempo medio necesario (10 minutos) se ha considerado que el valor de esta variable puede aportar 4 de 6 aciertos, es decir, el coeficiente c tendría valor 0.4 ($0.4 \cdot 10 = 4$).

El reparto de los otros dos aciertos, considerando que las navegaciones y las revisiones pueden llegar a tener una repercusión similar, se van a repartir equitativamente entre estas. Por un lado, el número medio de navegaciones necesario aportará 1 de 6 aciertos y por consiguiente d valdrá 0.5 ($0.5 \cdot 2 = 1$). Por otro lado, sabiendo que el contenido de la primera lección se implica en dos de las seis cuestiones del test y el de la segunda lección en cuatro de las seis cuestiones, el acierto restante se va a repartir de manera proporcional entre las variables de revisión de estas lecciones. Es decir, el coeficiente a valdrá 0.033 y el coeficiente b valdrá 0.067 ($0.033 \cdot 10 + 0.067 \cdot 10 = 1$).

En resumen, los coeficientes deducidos a partir del escenario ideal darían el siguiente resultado:

$$6 = 0.033 \cdot 10 + 0.067 \cdot 10 + 0.4 \cdot 10 + 0.5 \cdot 2 \qquad (3)$$

Una vez establecidos los coeficientes de cada una de las variables, y una vez conocidos los valores medios de cada una de ellas en el escenario ideal, se puede llevar a cabo el proceso de simulación. Concretamente, se van a generar valores aleatorios para cada una de las variables explicativas siguiendo una distribución normal, donde se va a tomar como media el valor del escenario ideal y una desviación típica acorde a la escala de la variable en cuestión, generando así diferentes tipos de progresos.

Con esto tendríamos generados para cada variable unos valores aleatorios acordes a las características del entorno real de los datos. Sin embargo, hay que tener en cuenta un detalle respecto a las variables de revisiones (x_1 y x_2), ya que la generación aleatoria tal como la hemos definido hasta ahora puede dar situaciones donde la variable del tiempo invertido (x_3) tenga un valor muy bajo, por ejemplo 1 minuto, y las variables de las revisiones tengan valores muy altos. Esto sería totalmente contradictorio, puesto que se necesita un mínimo de tiempo invertido para que la generación de eventos de revisión tenga lugar. Es decir, el

Tabla 5. Seis primeras filas del conjunto de datos generado para el Test 1

tiempo	navegaciones	s02l01	s02l02	aciertos
2	2	2	0	2
6	1	2	4	3
8	0	4	3	4
1	2	0	1	1
8	3	1	2	5
7	2	2	4	4

tiempo y las revisiones tienen una relación directamente proporcional y hay que tener en cuenta este aspecto en la generación de los datos.

Para ello, en el proceso de la generación aleatoria de las revisiones se toma como valor medio la mitad del tiempo generado y como desviación típica la cuarta parte de este tiempo. De esta manera, hacemos que la generación de los valores de las variables de revisiones y tiempo guarden una relación lógica entre ellas.

Por otro lado, debido a la desviación típica, se pueden generar valores negativos para algunas de las variables y en dichos casos se hace un filtrado para establecer a cero todos los valores en los que se de esta situación.

Una vez generados los valores de cada una de las variables explicativas, el siguiente paso es obtener los valores de la variable respuesta (y) mediante la sustitución de dichos valores en la ecuación definida anteriormente, multiplicándolos por sus correspondientes coeficientes y dando como resultado el número de aciertos obtenidos a partir del progreso simulado.

Por último, debido a que el valor de aciertos es un número entero y positivo, se hace un proceso de redondeo para los valores decimales obtenidos y se establecen a cero aquellos resultados negativos. Por otro lado, se puede dar el caso donde los valores del progreso sean mayores respecto a los del escenario ideal, lo cual representaría el caso de un estudiante que ha estudiado o se ha preparado con un progreso por encima de la media. Ante esta situación la ecuación daría lugar a un valor de aciertos mayor que 6, lo cual supera el límite posible y, por lo tanto, se filtran estos casos para reducir el valor de aciertos al

límite posible. En la Tabla 5 se puede ver una muestra del conjunto de datos sintéticos generados para el Test 1.

7.2.2. Datos del Test 2

Las variables que representan el progreso necesario para la predicción del rendimiento en el Test 2 son las siguientes:

- Variables explicativas:
  - s02l04: número de revisiones en la primera lección implicada en el test. Esta variable de aquí en adelante será representada como x_1.
  - s02l05: número de revisiones en la segunda lección implicada en el test. Esta variable de aquí en adelante será representada como x_2.
  - s02l06: número de revisiones en la tercera lección implicada en el test. Esta variable de aquí en adelante será representada como x_3.
  - tiempo: número de minutos invertidos en las lecciones y en la realización del test. Esta variable de aquí en adelante será representada como x_4.
  - navegaciones: número de navegaciones a cualquiera de las lecciones implicadas en el test. Esta variable de aquí en adelante será representada como x_5.
- Variable respuesta:
  - aciertos: número de cuestiones acertadas de las cinco planteadas en el test. Esta variable de aquí en adelante será representada como y.

Siguiendo la misma estrategia definida en el apartado anterior, se puede representar la relación entre todas las variables de la siguiente manera:

$$y = a \cdot x_1 + b \cdot x_2 + c \cdot x_3 + d \cdot x_4 + e \cdot x_5 \quad (4)$$

La ecuación 4 representa la aportación de cada una de las variables explicativas (x_n) en el resultado final de la variable respuesta (y). La repercusión o peso de cada variable en el resultado final se medirá en función del valor de su correspondiente coeficiente ($a, b, c, d\ y\ e$) y, tal como se hizo en el apartado anterior, estos valores son fijos y los vamos

a establecer como autores del curso e-learning. No obstante, antes de ello hay que definir el escenario ideal necesario para obtener la máxima nota en el test. En este caso sería necesario un número medio de 5, 10 y 5 revisiones respectivamente para cada una de las lecciones (x_1, x_2 y x_3), 8 minutos como tiempo medio invertido (x_4) y una media de 3 navegaciones (x_5). Con estos valores la ecuación quedaría de la siguiente manera:

$$5 = a \cdot 5 + b \cdot 10 + c \cdot 5 + d \cdot 8 + e \cdot 3 \qquad (5)$$

Para establecer los coeficientes se van a seguir las mismas proporciones del reparto de aciertos aplicado entre las variables del apartado anterior. En cuanto a la variable de tiempo invertido, teniendo en cuenta que este test está formado por 5 cuestiones, si en el apartado anterior el valor ideal de esta variable proporcionaba 4 de 6 aciertos, en este caso el equivalente serían 3.32 aciertos de 5 y, por lo tanto, el coeficiente d va a tener valor 0.415 ($0.415 \cdot 8 = 3.32$).

El ratio de aciertos restante (1.68) se va a repartir de manera equitativa entre el grupo de variables de revisiones y la variable de navegaciones. Por un lado, el número medio ideal de navegaciones aportará 0.84 aciertos y, por lo tanto, el coeficiente e tendrá valor 0.28 ($0.28 \cdot 3 = 0.84$).

Por otro lado, el 0.84 de aciertos restante se va a repartir de manera proporcional entre las variables de las revisiones, en función de la implicación de cada una de las lecciones en las cuestiones del test. Concretamente, la primera lección se implica en una de las cinco cuestiones y las lecciones segunda y tercera se implican en dos de las cinco lecciones respectivamente. Por lo tanto, los coeficientes a, b y c van a tener valores 0.0336, 0.0336 y 0.0672 respectivamente ($0.0336 \cdot 5 + 0.0336 \cdot 10 + 0.0672 \cdot 5 = 0.168 + 0.336 + 0.336 = 0.84$).

En resumen, los coeficientes deducidos a partir del escenario ideal darían el siguiente resultado:

$$5 = 0.0336 \cdot 5 + 0.0336 \cdot 10 + 0.0672 \cdot 5 + 0.4150 \cdot 8 + 0.2800 \cdot 3 \quad (6)$$

Una vez establecidos los coeficientes de cada una de las variables, y una vez conocidos los valores medios de cada una de ellas en el escenario ideal, se ha llevado a cabo el proceso de simulación de la misma manera que se ha explicado en el apartado anterior, obteniendo así un conjunto de datos cuya muestra se puede ver en la Tabla 6.

Tabla 6. Seis primeras filas del conjunto de datos generado para el Test 2

tiempo	navegaciones	s02l04	s02l05	s02l06	aciertos
2	0	2	1	0	1
5	2	4	7	2	3
7	3	6	0	6	4
1	0	0	0	1	0
7	2	1	0	0	3
6	0	2	8	3	3

7.2.3. Datos del Test 3

Las variables que representan el progreso necesario para la predicción del rendimiento en el Test 3 son las siguientes:

- Variables explicativas:
  - s03l01: número de revisiones en la primera lección implicada en el test. Esta variable de aquí en adelante será representada como x_1.
  - s03l02: número de revisiones en la segunda lección implicada en el test. Esta variable de aquí en adelante será representada como x_2.
  - s03l03: número de revisiones en la tercera lección implicada en el test. Esta variable de aquí en adelante será representada como x_3.
  - s03l05: número de revisiones en la cuarta lección implicada en el test. Esta variable de aquí en adelante será representada como x_4.
  - s03l06: número de revisiones en la quinta lección implicada en el test. Esta variable de aquí en adelante será representada como x_5.
  - tiempo: número de minutos invertidos en las lecciones y en la realización del test. Esta variable de aquí en adelante será representada como x_6.
  - navegaciones: número de navegaciones a cualquiera de las lecciones implicadas en el test. Esta variable de aquí en adelante será representada como x_7.

- Variable respuesta:
    - aciertos: número de cuestiones acertadas de las cinco planteadas en el test. Esta variable de aquí en adelante será representada como y.

Siguiendo la misma estrategia de los apartados anteriores, se puede representar la relación entre todas las variables de la siguiente manera:

$$y = a \cdot x_1 + b \cdot x_2 + c \cdot x_3 + d \cdot x_4 + e \cdot x_5 + f \cdot x_6 + g \cdot x_7 \quad (7)$$

La ecuación 7 representa la aportación de cada una de las variables explicativas (x_n) en el resultado final de la variable respuesta ($\boldsymbol{y}$). La repercusión o peso de cada variable en el resultado final se medirá en función del valor de su correspondiente coeficiente ($a, b, c, d, e, f\ y\ g$) y, tal como se hizo en los apartados anteriores, estos valores son fijos y los vamos a establecer como autores del curso e-learning. No obstante, antes de ello hay que definir el escenario ideal necesario para obtener la máxima nota en el test. En este caso sería necesario un número medio de 15 revisiones para cada una de las lecciones (x_1, x_2, x_3, x_4 y x_5), 20 minutos como tiempo medio invertido (x_6) y una media de 5 navegaciones (x_7). Con estos valores la ecuación quedaría de la siguiente manera:

$$5 = a \cdot 15 + b \cdot 15 + c \cdot 15 + d \cdot 15 + e \cdot 15 + f \cdot 20 + g \cdot 5 \quad (8)$$

Para establecer los coeficientes se van a seguir las mismas proporciones del reparto de aciertos aplicado entre las variables de los apartados anteriores. En cuanto a la variable de tiempo invertido, teniendo en cuenta que este test está formado por 5 cuestiones, al igual que en el apartado anterior, el equivalente son 3.32 aciertos de 5. Por lo tanto, el coeficiente f va a tener valor 0.166 ($0.166 \cdot 20 = 3.32$).

El ratio de aciertos restante (1.68) se va a repartir de manera equitativa entre el grupo de variables de revisiones y la variable de navegaciones. Por un lado, el número medio ideal de navegaciones aportará 0.84 aciertos y, por lo tanto, el coeficiente g tendrá valor 0.168 ($0.168 \cdot 5 = 0.84$). Por otro lado, el 0.84 de aciertos restante se va a repartir de manera proporcional entre las variables de las revisiones, en función de la implicación de cada una de las lecciones en las cuestiones del test. En este caso, para cada una de las cinco cuestiones se implica una lección diferente, por lo que tendrán la misma proporción. Es decir, los coeficientes $a, b, c, d\ y\ e$ van a tener valor 0.0112 ($0.0112 \cdot 15 \cdot 5 = 0.84$).

Tabla 7. Seis primeras filas del conjunto de datos generado para el Test 3

tiempo	navegaciones	s03l01	s03l02	s03l03	s03l05	s03l06	aciertos
7	0	6	3	0	7	10	1
13	4	4	13	0	14	5	3
17	6	0	9	0	1	0	4
6	0	2	8	3	6	5	1
17	4	17	2	25	12	3	4
15	1	0	11	8	7	0	3

En resumen, los coeficientes deducidos a partir del escenario ideal darían el siguiente resultado:

$$5 = 0.0112 \cdot 15 + 0.0112 \cdot 15 + 0.0112 \cdot 15 + 0.0112 \cdot 15 + 0.0112 \cdot 15 + 0.166 \cdot 20 + 0.168 \cdot 5 \quad (9)$$

Una vez establecidos los coeficientes de cada una de las variables, y una vez conocidos los valores medios de cada una de ellas en el escenario ideal, se ha llevado a cabo el proceso de simulación al igual que en los apartados anteriores, obteniendo así un conjunto de datos cuya muestra se puede ver en la Tabla 7.

8. Desarrollo e implementación del modelo predictivo

Tras completar las fases de análisis, procesamiento y generación de datos estamos preparados para proceder con la implementación del modelo predictivo. En este apartado vamos a implementar los tres modelos encargados de predecir el rendimiento para cada uno de los tests que conforman el curso e-learning. Estos modelos son independientes debido a las particularidades de los datos que representan el progreso implicado en la realización de cada uno de dichos tests.

Existen diferentes métodos y técnicas para implementar sistemas predictivos mediante aprendizaje automático, pero en este caso aplicaremos dos de ellas. Por un lado, se empleará el método de regresión lineal debido a la naturaleza de nuestros datos, los cuales han sido descritos a partir de ecuaciones lineales y, por lo tanto, este método puede ser una herramienta decisiva para nuestra solución. Por otro lado, gracias al auge alcanzado en los últimos años y el gran rendimiento que pueden proporcionar, se hará uso de redes neuronales para comprobar si existe una mejora en el rendimiento respecto al método de regresión lineal.

8.1. Regresión lineal

A pesar de su sencillez, el modelo de regresión lineal sigue siendo una herramienta muy utilizada en el aprendizaje estadístico. Muchos de los métodos de aprendizaje automático supervisado son generalizaciones del modelo de regresión lineal [57].

En un problema de regresión lineal tenemos una población objeto de estudio P de tamaño $|P|$, una variable respuesta cuantitativa Y, un vector X de p variables explicativas $(X_1, X_2, \dots, X_p)$ y un conjunto de datos de entrenamiento D_{train} de tamaño n_{train} donde:

$$D_{train} = (x_1, y_1), \dots, \left(x_{n_{train}}, y_{n_{train}}\right),\ x_i = \left(x_{i1}, x_{i2}, \dots, x_{ip}\right)$$

Teniendo estos ingredientes el objetivo es determinar una función f capaz de predecir la variable respuesta (Y) a partir de un valor observado de las variables explicativas (X),

pudiendo encontrar y explicar la relación entre X e Y. Para ello se asume que la función de regresión es lineal o puede ser aproximada por una función lineal, donde el modelo resultante es el siguiente:

$$Y = \beta_0 + \beta_1 X_1 + \beta_1 X_2 + \cdots + \beta_p X_p + \epsilon \qquad (10),$$

donde $\beta_0, \beta_1, \ldots, \beta_p$ son $p + 1$ constantes que representan los coeficientes o parámetros del modelo, y ϵ es un término de error. Por lo tanto, el problema de la estimación de la función f se reduce al problema de estimar el vector de parámetros $\beta = (\beta_0, \ldots, \beta_p)$. Este vector será estimado a partir de los datos de entrenamiento y un método de estimación, denotándolo como $\hat{\beta} = (\hat{\beta}_0, \ldots, \hat{\beta}_p)$ y la correspondiente función de regresión estimada será:

$$\hat{f}(x) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p \qquad (11)$$

El método de estimación más utilizado es el de estimación por mínimos cuadrados o least squares, mediante el cual se obtiene el vector $\hat{\beta}$ como solución al siguiente problema [58]:

$$\min_{(\beta_0,\ldots,\beta_p) \in \mathbb{R}^{p+1}} \sum_{i=1}^{n_{train}} \left[y_i - (\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) \right]^2 \qquad (12)$$

Resolver el problema de optimización anterior es equivalente a definir el estimador de la función de regresión como la función que minimiza el error de entrenamiento, es decir, la que mejor se ajusta a los datos.

Por otro lado, la predicción de la variable respuesta para el i-ésimo individuo en D_{train} con su correspondiente vector de variables explicativas se denota como $\hat{y}_i$, el residuo i-ésimo (ϵ_i) se define como: $\hat{\epsilon}_i = y_i - \hat{y}_i$, y la suma de los cuadrados de los residuos se denota como RSS o Residual Sum of Squares:

$$\text{RSS} = \sum_{i=1}^{n_{train}} \hat{\epsilon}_i^2 = \sum_{i=1}^{n_{train}} \left[y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_p x_{ip}) \right]^2 = \sum_{i=1}^{n_{train}} [y_i - \hat{y}_i]^2 \qquad (13)$$

Por lo tanto, de acuerdo con la ecuación 12, el estimador por mínimos cuadrados del vector de parámetros β es el que minimiza la suma de los cuadrados de los residuos,

$$(\hat{\beta}_0, \ldots, \hat{\beta}_1) = \text{argmin}_{(\beta_0,\ldots,\beta_p) \in \mathbb{R}^{p+1}} \sum_{i=1}^{n_{train}} \hat{\epsilon}_i^2 \qquad (14)$$

Una medida del error de predicción de $\hat{f}(x)$ es el error de generalización o Mean Square Error (MSE), que se define como el riesgo de $\hat{f}(x)$:

$$R(\hat{f}) = E\left[L\left(\hat{f}(\boldsymbol{X}) - Y\right)\right] = \frac{\sum_{(x,y)\epsilon P} L\left(y - \hat{f}(\boldsymbol{x})\right)}{|P|} = \mathrm{E}\left[\left(\hat{\mathrm{f}}(\mathbf{X}) - \mathrm{Y}\right)^2\right] =$$

$$= \frac{\sum_{(x,y)\epsilon P}\left[y - \hat{f}(\boldsymbol{x})\right]^2}{|P|} \equiv MSE\left(\hat{f}\right) \quad (15),$$

donde P es el conjunto de parejas $(\boldsymbol{x}, y)$ en la población de datos y $|P|$ es el cardinal de P. Es decir, para cada $\boldsymbol{x}$, $\hat{\mathrm{f}}(\mathbf{x})$ es un número real. Intuitivamente se podría considerar como aproximación del error de generalización de $\hat{f}(\boldsymbol{x})$ la pérdida esperada del estimador en el conjunto de datos de entrenamiento, lo cual se conoce como error de entrenamiento o riesgo empírico (Err_{train}). Sin embargo, este error proporciona una buena estimación del ajuste del modelo estimado $\hat{\mathrm{f}}(\mathrm{x})$ a los datos de entrenamiento, pero una estimación muy pobre del error de generalización. Dada una clase de funciones de predicción o modelos ($M = \{M_1, \dots, M_m\}$), la función de predicción óptima es la que tiene menor error de generalización.

Por otro lado, un aspecto fundamental en los problemas de regresión es la multicolinealidad. Es decir, la correlación entre las variables explicativas, lo cual complica notablemente la interpretación de los resultados de un modelo de regresión. La presencia de interacciones entre variables explicativas, si no se modela correctamente, puede inducir a interpretaciones incorrectas de los coeficientes de regresión ya que no resulta posible separar los efectos de cada variable explicativa y medir la contribución individual de cada una de ellas [59].

Para detectar la multicolinealidad se puede hacer uso de la matriz de correlación de las variables explicativas, donde cualquier alto valor de correlación será un indicio de multicolinealidad. Por otro lado, se puede hacer uso del factor de inflación de la varianza o Variance Inflation Factor (VIF). Con este indicador si obtenemos un valor mayor o igual que 5 significará que la multicolinealidad es un problema, y si se obtiene un valor igual a 1 significa que hay ausencia de multicolinealidad.

Es posible que exista multicolinealidad aun cuando todos los valores de la matriz de correlación de las variables explicativas sean bajos. Esto ocurre cuando un regresor es aproximadamente una combinación lineal de otras variables explicativas. En dichos casos el factor VIF permite detectar mejor la existencia de la multicolinealidad. Cuando se detecta multicolinealidad se puede eliminar del modelo de regresión una o varias de las variables explicativas responsables de dicha multicolinealidad. El impacto en el ajuste global del

modelo debería ser pequeño ya que, si dos variables están correlacionadas y eliminamos una de ellas, la información de la variable eliminada ya está contenida de cierta manera en la variable mantenida.

Para la implementación de cada uno de nuestros modelos nos vamos a basar en estudiar la multicolinealidad y la importancia de cada variable explicativa en la predicción de la variable respuesta y, por otro lado, se va a hacer uso de métodos de selección de variables. Estos métodos permiten escoger el mejor conjunto de variables en relación con su aportación en el rendimiento del modelo, permitiendo evitar el ajuste de modelos innecesariamente complejos con el objetivo de seleccionar modelos más sencillos que explican igual de bien los datos observados y son más fácilmente interpretables. Es decir, dado un modelo de regresión lineal con p regresores que llamaremos "modelo completo", en muchos casos es conveniente considerar modelos más sencillos o, lo que es lo mismo, menos flexibles. Cuantos más regresores contenga el modelo, matemáticamente será más flexible en su adaptación a la tendencia de los datos.

Una flexibilidad excesiva conduce a lo que se conoce como sobreajuste u overfiting, que consiste en ajustar a los datos un modelo innecesariamente complejo hasta el extremo en el que este consigue calcar el ruido de los datos. Este tipo de modelos obtienen errores de entrenamiento muy bajos, pero sin embargo tienen una muy baja capacidad de generalización obteniendo así unas muy malas predicciones sobre datos que no han aprendido con antelación.

Los métodos de selección de variables más utilizados son los siguientes [60]:

- Método del mejor subconjunto posible o best subset selection: consiste en formar todos los posibles subconjuntos de variables explicativas (2^p combinaciones), ajustar cada uno de dichos modelos de regresión y finalmente se elige el mejor modelo entre los 2^p candidatos de acuerdo con un criterio de selección de modelos. La limitación de este método es que el número de modelos a considerar aumenta rápidamente con el número de variables explicativas p. Por ejemplo, para $p = 20$ hay $2^{20} = 1048576$ modelos posibles. En la práctica el algoritmo deja de ser viable si $p > 40$.

- Métodos secuenciales o de paso: cuando una búsqueda exhaustiva de todos los submodelos posibles no es viable, una alternativa razonable consiste en explorar el espacio de los posibles modelos considerando "un buen camino" a través de dicho espacio. Los métodos secuenciales se basan en esta estrategia, considerándose como algoritmos voraces o greedy donde se reemplaza la búsqueda de un óptimo global por la consideración sucesiva de óptimos locales. Con lo cual no garantizan la mejor solución, pero cuando no es viable llevar a cabo el método del mejor subconjunto son una muy buena alternativa.

  Dentro de los métodos secuenciales los más empleados son el método de selección hacia delante o forward stepwise selection y el método de selección hacia atrás o backward stepwise selection. En el primero se considera como punto de partida el modelo nulo, es decir, aquel que no tiene predictores y se va agregando en cada paso un nuevo predictor hasta que todos los predictores son incluidos en el modelo. En particular, en cada paso se agrega al modelo la variable explicativa que proporciona la mayor mejora adicional al juste global del modelo.

  En el segundo método se sigue la misma filosofía, pero el punto de partida es el modelo completo, es decir, aquel que contiene todos los predictores. En cada paso se va eliminando la variable explicativa cuya exclusión proporciona la menor disminución del ajuste global del modelo.

  De esta manera, con estos algoritmos voraces se construye un total de $p + 1$ modelos, y finalmente se selecciona el mejor de ellos de acuerdo con un criterio de selección de modelos.

En nuestra implementación vamos a hacer uso del método del mejor subconjunto, ya que el mayor número de variables explicativas que tenemos es 7 en el modelo del Test 3, lo cual es una cantidad de combinaciones ($2^7 = 128$) asumible asegurándonos de probar todas las combinaciones posibles.

Como se ha mencionado en la explicación de estos métodos, tras generar el conjunto de modelos hay que seleccionar el mejor de ellos según un criterio de selección.

Concretamente vamos a emplear los criterios más utilizados, los cuales son los cinco siguientes:

- C_p de Mallows, definido como:

$$C_p(M) = \frac{1}{n_{train}}[RSS(M) + 2\hat{\sigma}^2 d] = \mathrm{Err}_{train}(M) + \frac{2\hat{\sigma}^2}{n_{train}}\mathrm{d},$$

donde $RSS(M)$ y $Err_{train}(M)$ representan la suma de los cuadrados de los residuos y el error de entrenamiento asociados al modelo M respectivamente, d es el número de parámetros y $\hat{\sigma}^2$ es una estimación de la varianza del término de error. Cuanto menor sea el valor de este coeficiente mejor será el modelo aproximado.

- Criterio de Información Bayesiano o Bayesian Information Criterion (BIC), definido como:

$$BIC(M) = -2logL_M + log(n_{train})d,$$

donde L_M es el valor máximo de la función de verosimilitud para el modelo M y d es el número de parámetros. Cuanto menor sea el valor de este coeficiente mejor será el modelo aproximado.

- Coeficiente de determinación ajustado (R^2_{adj}), definido como:

$$R^2_{adj}(M) = 1 - \frac{RSS(M)/(n_{train}-(d+1))}{TSS/(n-1)} = 1 - (1-R^2)\frac{(n-1)}{n_{train}-(d+1)},$$

donde $RSS(M)$ representa la suma de los cuadrados de los residuos asociada al modelo M, $TSS\ (Total\ Sum\ of\ Squares) = \sum_{i=1}^{n_{train}}(y_i - \bar{y})^2$ y d es el número de parámetros. Cuanto mayor sea el valor de este coeficiente mejor será el modelo aproximado.

- División de datos o Data split: si el tamaño muestral de los datos es lo suficientemente grande se considera una partición de estos en dos subconjuntos D_{train} y D_{test}. Sea $\hat{f}_{train}(.)$ la estimación de $f(.)$ obtenida a partir de D_{train}, se estima el error de generalización como el error cuadrático medio en el conjunto D_{test} de manera que cuanto menor sea este error mejor será el modelo aproximado. Se recomienda repetir el procedimiento N veces y dar como estimación del error de generalización la media de los N valores obtenidos.
- Validación cruzada o Cross Validation (CV): se considera una partición aleatoria de los datos iniciales en K subconjuntos y se llevan a cabo K iteraciones. En cada iteración se utiliza uno de los subconjuntos como conjunto de datos de validación y los datos relativos a los restantes subconjuntos se utilizan como conjunto de datos de entrenamiento. Finalmente, se calcula la suma del error cuadrático de predicción en el conjunto de validación y se proporciona como estimación del error de

generalización el promedio de las K estimaciones. Cuanto menor sea la estimación del error de generalización mejor será el modelo aproximado.

La implementación de los modelos de este apartado será llevada a cabo mediante el lenguaje de programación R.

8.1.1. Test 1

El conjunto de datos del Test 1 está formado por cuatro variables explicativas ("tiempo", "navegaciones", "s02l01" y "s02l02") y la variable respuesta ("aciertos"). Si calculamos la matriz de correlación y la representamos gráficamente, como en la Figura 19, se puede observar que la variable "tiempo" tiene un alto valor de correlación con las variables "s02l01" y "s02l02" (revisiones de las lecciones 1 y 2). Está correlación refleja la relación que hay entre el tiempo invertido y el número de revisiones realizadas, tal como se determinó en la generación de los datos.

A parte de esta correlación, entre el resto de las variables explicativas no se detecta un caso tan significativo como este último. Por otro lado, las variables que mejor explican la variable respuesta, por su correlación con esta, son las de tiempo y revisiones. Es lógico que las variables de tiempo y revisiones expliquen en una escala similar la variable respuesta, debido a la gran correlación que hay entre estas variables explicativas. Es decir, la información que contiene una de ellas de alguna manera también está contenida en el resto, y si una de ellas explica de manera significativa la variable respuesta las otras también lo harán.

En la Figura 20 se ha representado el coeficiente VIF para cada una de las variables explicativas o regresores para corroborar los casos de correlación detectados en la matriz anterior. Como se puede ver, la variable "navegaciones" tiene una nula correlación y, por otro lado, las variables "tiempo", "s02l01" y "s02l02" tienen un valor mayor. Es decir, se confirman los casos de correlación antes detectados. Sin embargo, no se supera en ningún caso el valor 5, por lo que no tenemos un problema de correlación extremadamente acentuado.

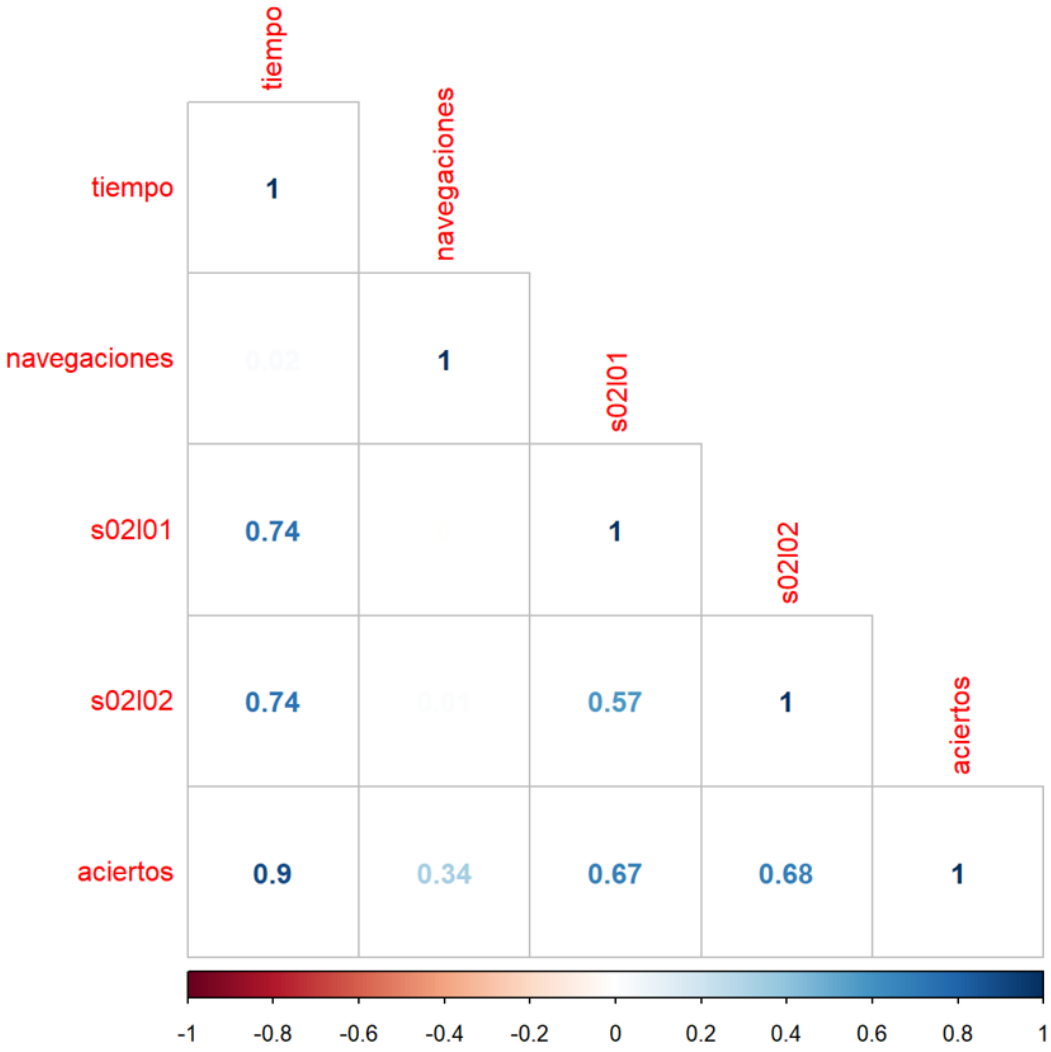

Figura 19. Correlación entre las variables del conjunto de datos del Test 1

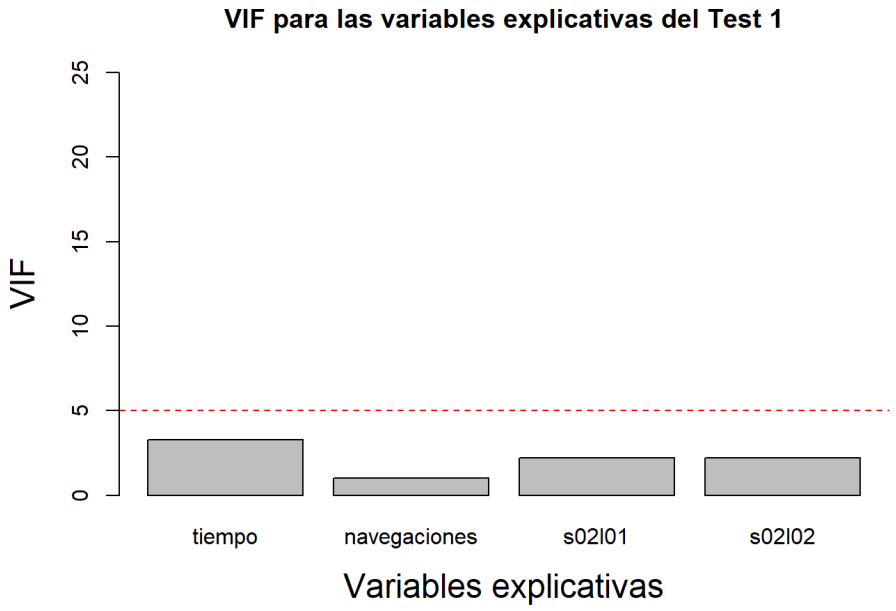

Figura 20. Factor de inflación de la varianza (VIF) para los regresores del Test 1

Tabla 8. Mejor combinación de variables del Test 1 en función del límite establecido

Nº de variables	tiempo	navegaciones	s02l01	s02l02
1	x			
2	x	x		
3	x	x	x	
4	x	x	x	x

Una vez analizada la relación entre las diferentes variables que componen el conjunto de datos del Test 1 el siguiente paso es comenzar con la implementación del modelo de regresión lineal. Para ello, se va a seguir la estrategia explicada en la introducción de este apartado; mediante el método de selección del mejor subconjunto se van a ajustar todas las combinaciones posibles seleccionando finalmente la mejor de ellas a partir de los criterios de selección planteados.

En la Tabla 8 se pueden ver las mejores combinaciones seleccionadas por el método del mejor subconjunto en función del número de variables explicativas tenidas en cuenta. Si se tiene en cuenta una única variable, la que mejor rendimiento aporta es la variable "tiempo" ya que, tal como se vio en la matriz de correlación, era la que mejor explicaba la variable respuesta. En caso de tener en cuenta dos variables se añade la variable "navegaciones". En este caso no se escoge ninguna de las variables de revisiones, ya que como vimos tienen una alta correlación con la variable "tiempo" y añadirlas seguramente no aportaría mucha más información que la contenida en dicha variable. Por último, si se quieren tener en cuenta tres y cuatro variables se añaden "s02l01" y "s02l02" respectivamente.

La Figura 21 muestra el error de entrenamiento obtenido con cada uno de los modelos y, como es de esperar, a medida que aumenta el número de variables el modelo se vuelve más flexible y este error va disminuyendo. El modelo con menor error de entrenamiento ha sido el que contiene las cuatro variables, es decir, el modelo completo. Sin embargo, este no es un indicador fiable para la selección del mejor modelo y por ello vamos a hacer uso de los criterios de selección anteriormente explicados, los cuales se basan en la estimación del error de generalización.

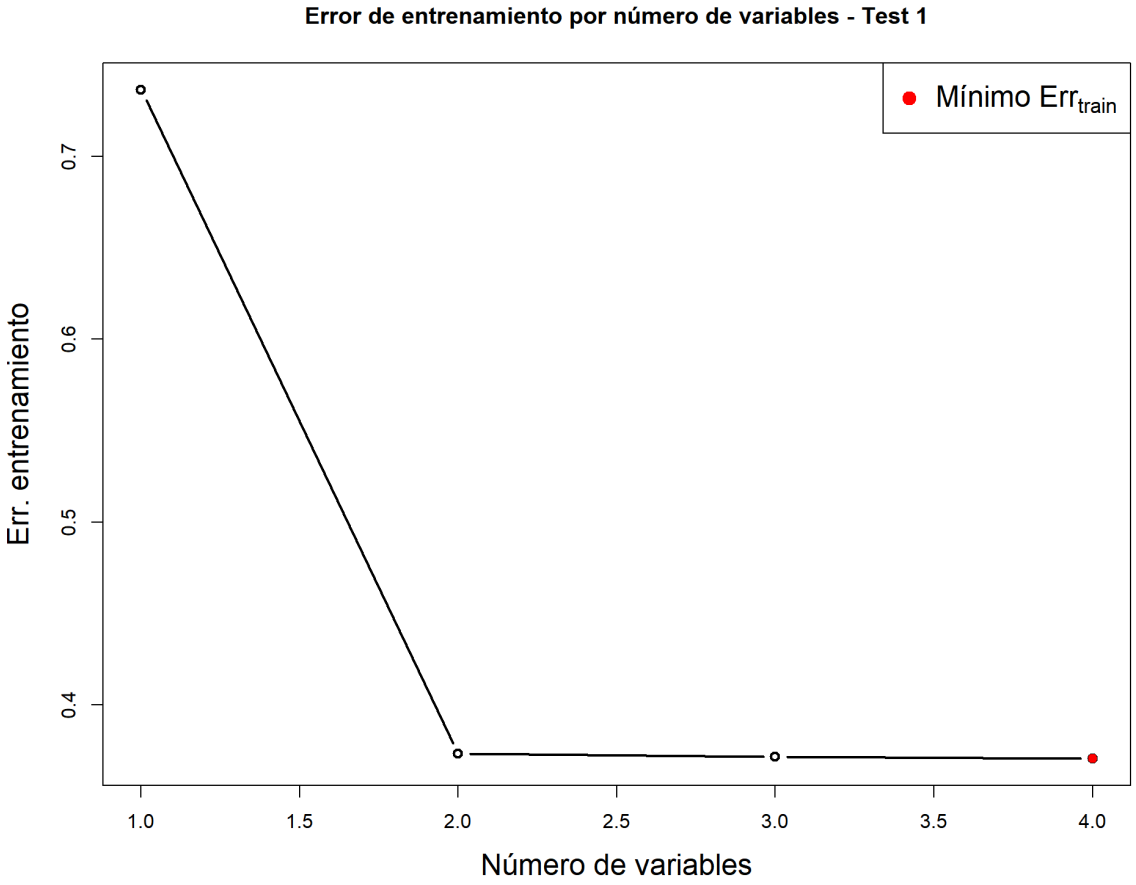

Figura 21. Error de entrenamiento de los modelos de regresión ajustados para el Test 1

Tabla 9. Errores de entrenamiento y generalización de los mejores modelos de regresión seleccionados para el Test 1

Núm. de variables	Error de entrenamiento	MSE
2	0.3733	0.2177
4	0.3708	0.2150

Los resultados de la selección según cada criterio están reflejados en la Figura 22, donde se puede ver en cada caso el mejor modelo marcado con un punto rojo. Como se puede ver, tres de los cinco criterios (C_p, R^2_{adj} y CV) han seleccionado el modelo de cuatro variables o modelo completo y, por otro lado, dos de ellos (BIC y $Data\ split$) han seleccionado el modelo con dos variables. Ante esta disparidad de selección, y para saber cuál de las dos opciones es más conveniente, se va a revisar el error de generalización o MSE de los modelos estimado sobre el conjunto de datos de prueba que se ha reservado al inicio de la implementación (25% de los datos). Como se puede ver en la Tabla 9, ambos modelos tienen un valor de MSE prácticamente igual, al igual que ocurre con el error de entrenamiento.

Criterios de selección del modelo del Test 1

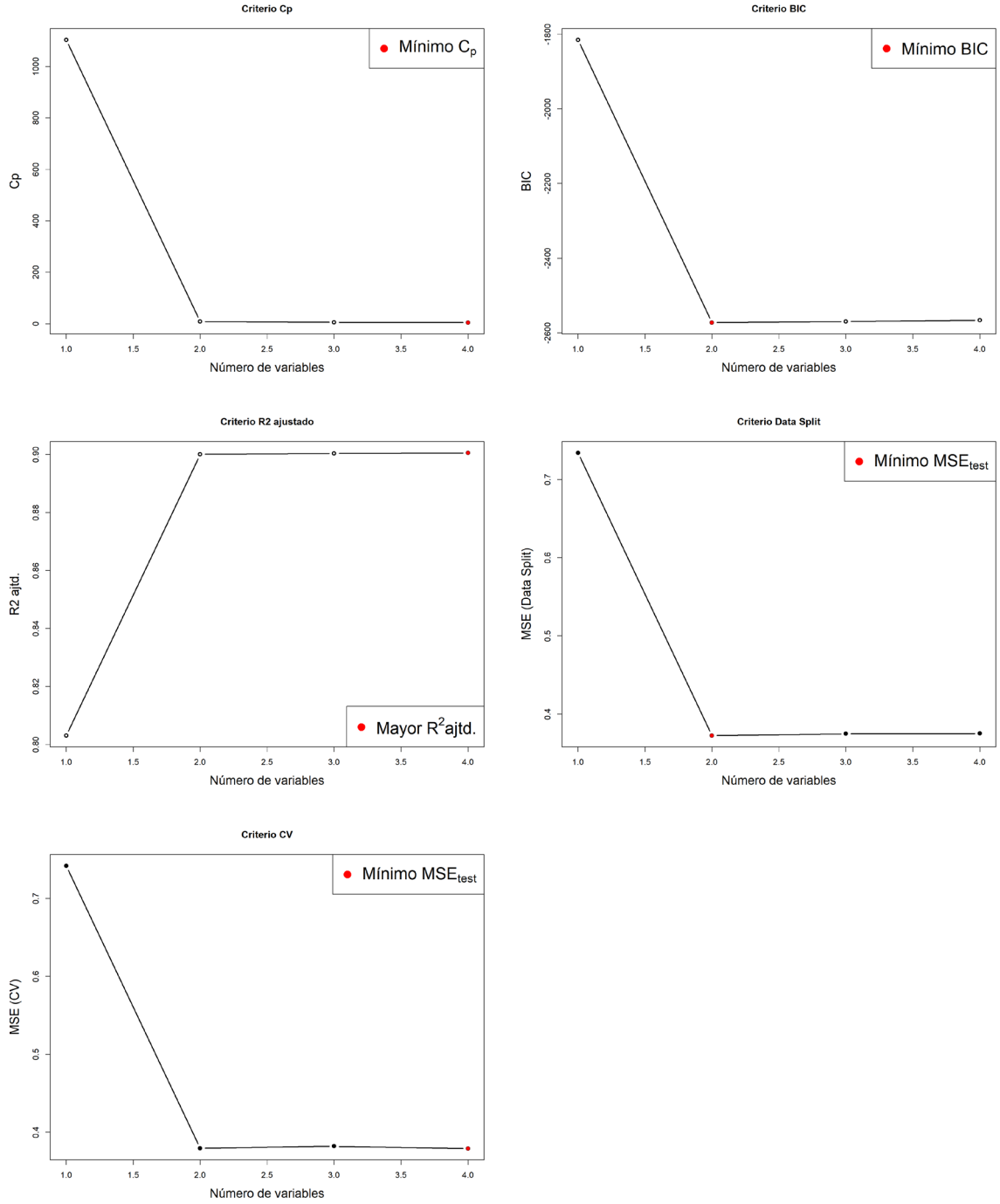

Figura 22. Modelo seleccionado para el Test 1 en función de cada criterio

Tabla 10. Coeficientes de regresión estimados para las variables del modelo del Test 1

Term. independiente	tiempo	navegaciones
0.4579	0.3673	0.3981

Tabla 11. Muestra de 20 predicciones realizadas sobre los datos de prueba del Test 1

Rendimiento predicho	Rendimiento verdadero
6.00	6
0.86	0
4.87	5
4.22	5
5.69	6
3.83	4
3.52	4
5.97	6
2.29	2
2.36	2
3.40	4
1.56	2
2.36	2
4.16	5
0.46	0
3.40	4
4.25	5
5.97	6
3.76	4
5.63	6

Por lo tanto, teniendo en cuenta que ambos modelos ofrecen un rendimiento casi idéntico, la mejor opción es el modelo más sencillo debido a su mejor interpretabilidad. Es decir, se va a seleccionar el modelo de dos variables, cuyos coeficientes se pueden ver en la Tabla 10. Respecto al MSE obtenido para este modelo, teniendo en cuenta que los valores de las predicciones pueden estar en el intervalo [0,6], un error con valor 0.2177 se puede considerar un muy buen resultado en dicha escala.

En total se han predicho 500 rendimientos y en la Tabla 11 se pueden ver los resultados de las 20 primeras predicciones. La primera columna de esta tabla muestra el rendimiento o

aciertos predichos y la segunda columna muestra el rendimiento o aciertos reales del caso correspondiente. De esta manera podemos ver el contraste entre la capacidad de predicción que tiene el modelo respecto a la realidad. Como es de esperar, al ser un modelo de regresión es muy complicado dar con predicciones de valores enteros y por ello prácticamente todos los valores tienen decimales. No obstante, debido al bajo MSE del modelo, son unos resultados muy orientativos de cara al estudiante pudiendo considerarlos como la nota sobre 6 que obtendría en el Test 1 en función de su progreso actual. Por otro lado, debido a que el número de aciertos está acotado en el intervalo [0,6], todas aquellas predicciones que se hayan salido de este rango han sido ajustadas a dichos límites.

8.1.2. Test 2

El conjunto de datos del Test 2 está formado por cinco variables explicativas ("tiempo", "navegaciones", "s02l04", "s02l05" y "s02l06") y la variable respuesta ("aciertos"). Si calculamos la matriz de correlación y la representamos gráficamente, como en la Figura 23, se puede observar que la variable "tiempo" tiene un alto valor de correlación con las tres variables de revisiones, siendo las más destacadas "s02l04" y "s02l06". Al igual que en el modelo anterior, esta correlación queda corroborada mediante el coeficiente VIF (Figura 24) y refleja la relación que hay entre el tiempo invertido y el número de revisiones realizadas, tal como se determinó en la generación de los datos.

Además de esta correlación, entre el resto de las variables explicativas no se detecta un caso tan significativo como este último. Por otro lado, tal como ocurría en el Test 1, las variables que mejor explican la variable respuesta son las de tiempo y revisiones, siendo no tan destacadas las navegaciones.

Siguiendo con la misma estrategia, tras analizar la relación entre las diferentes variables que componen el conjunto de datos del Test 2, el siguiente paso es comenzar con la implementación del modelo de regresión lineal. Mediante el método del mejor subconjunto se han buscado las mejores combinaciones de las variables para tener en cuenta en función del límite de estas, y el resultado se puede ver en la Tabla 12.

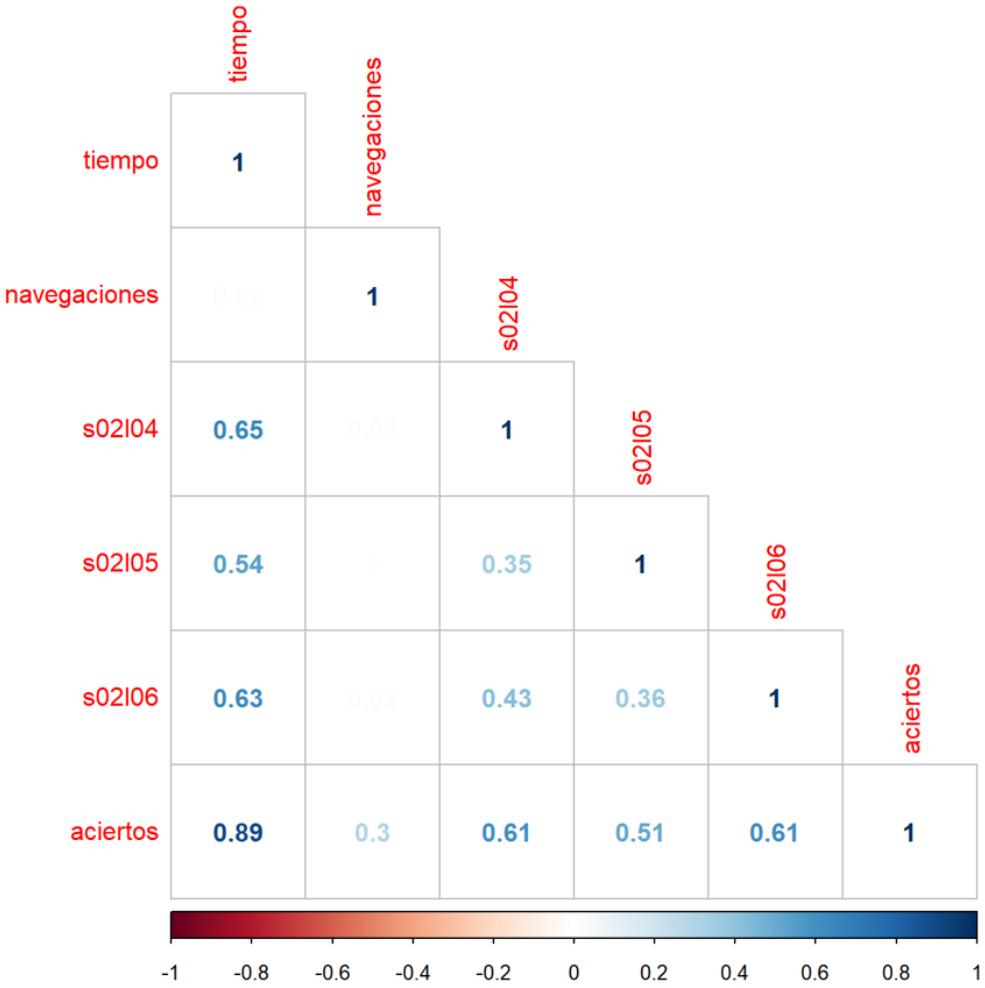

Figura 23. Correlación entre las variables del conjunto de datos del Test 2

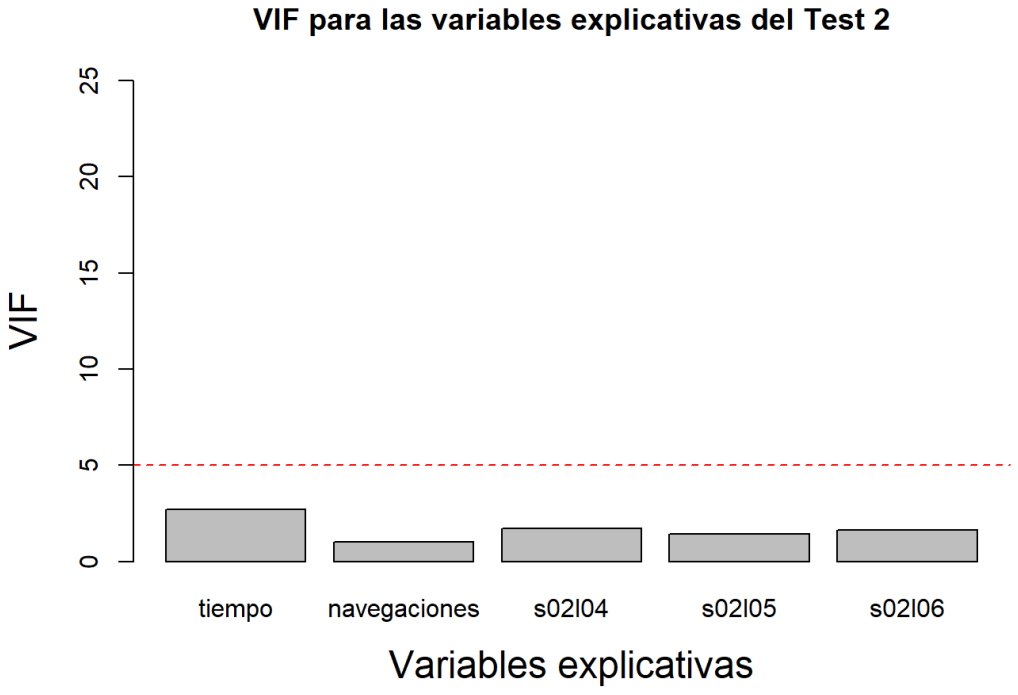

Figura 24. Factor de inflación de la varianza (VIF) para los regresores del Test 2

Tabla 12. Mejor combinación de variables del Test 2 en función del límite establecido

Nº de variables	tiempo	navegaciones	s02l04	s02l05	s02l06
1	x				
2	x	x			
3	x	x			x
4	x	x		x	x
5	x	x	x	x	x

Si se tiene en cuenta una única variable, la que mejor rendimiento aporta es la variable "tiempo" ya que, tal como se vio en la matriz de correlación, era la que mejor explicaba la variable respuesta. En caso de tener en cuenta dos variables se añade la variable "navegaciones". No se escoge ninguna de las variables de revisiones ya que tienen una alta correlación con la variable "tiempo" y añadirlas seguramente no aporte mucha más información que la contenida en dicha variable. Al tener en cuenta tres, cuatro y cinco variables explicativas comienzan a añadirse las variables de revisiones respectivamente.

La Figura 25 muestra el error de entrenamiento obtenido con cada uno de los modelos y, como se ha explicado en apartados anteriores, a medida que aumenta el número de variables el modelo se vuelve más flexible y este error va disminuyendo. Por lo tanto, el modelo con menor error de entrenamiento ha sido el modelo completo (5 variables).

Si revisamos la selección realizada por cada uno de los criterios en la Figura 26, por unanimidad, en este caso todos los criterios han seleccionado el modelo completo con las cinco variables explicativas. En la Tabla 13 se pueden ver los coeficientes estimados mediante regresión lineal para cada una de dichas variables junto al término independiente.

Para terminar con la implementación de este modelo, y como se hizo con el modelo del apartado anterior, se ha llevado a cabo la predicción sobre los datos de prueba o test que se han reservado al principio de la implementación (25% de los datos). En la Tabla 14 se puede ver el MSE estimado sobre dichos datos. Teniendo en cuenta que los valores de las predicciones pueden estar en el intervalo [0,5], un MSE con valor 0.1524 se puede considerar un muy buen resultado en dicha escala.

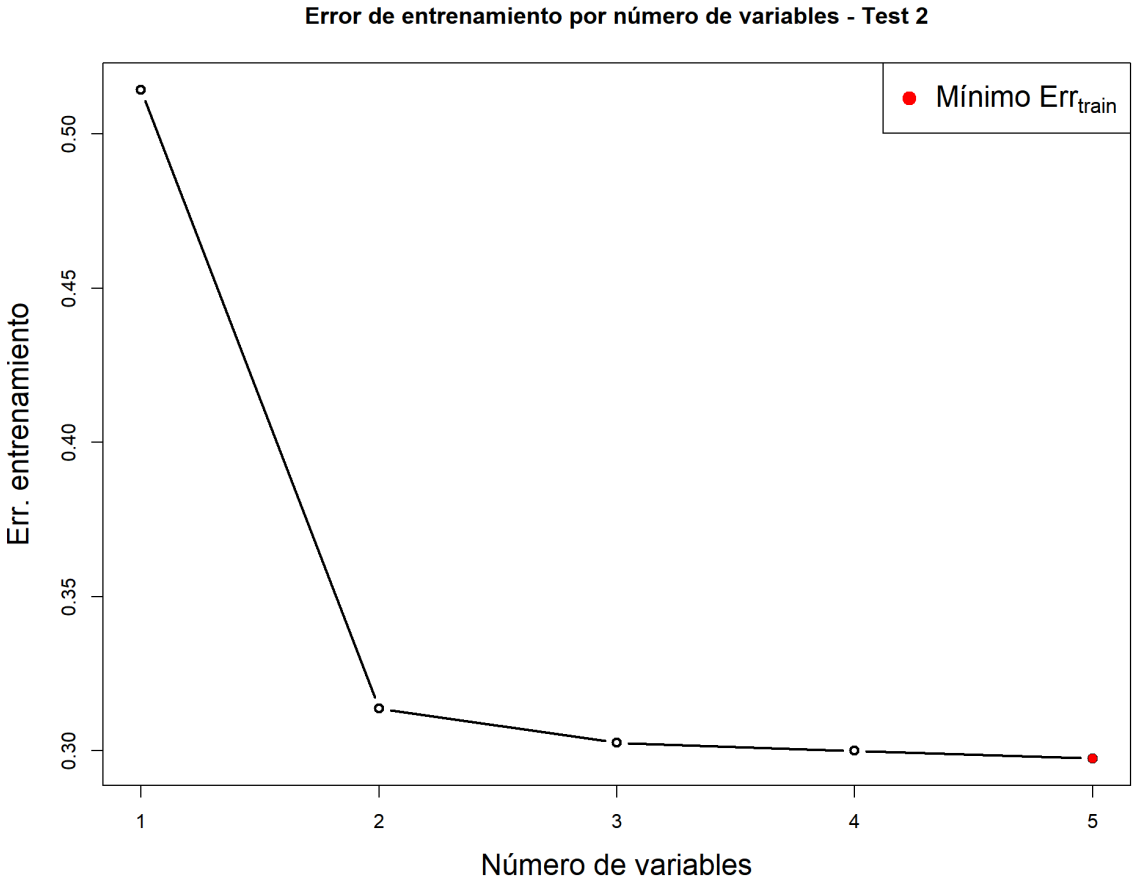

Figura 25. Error de entrenamiento de los modelos de regresión ajustados para el Test 2

Tabla 13. Coeficientes de regresión estimados para las variables del modelo del Test 2

Term. independiente	tiempo	navegaciones	s02l04	s02l05	s02l06
0.5362	0.3296	0.2171	0.0224	0.0167	0.0468

Tabla 14. Errores de entrenamiento y generalización estimados para el modelo de regresión del Test 2

Error de entrenamiento	MSE
0.2975	0.1524

Por otro lado, en la Tabla 15 se pueden ver 20 de las 500 predicciones, donde la primera columna contiene las predicciones del rendimiento realizadas por el modelo y la segunda columna contiene el rendimiento o aciertos verdaderos del caso correspondiente. Tal como ocurrió con el modelo del Test 1, viendo los resultados de predicción y teniendo en cuenta el MSE obtenido, se puede considerar que este modelo predice el rendimiento de los usuarios con unas aproximaciones muy buenas.

Criterios de selección del modelo del Test 2

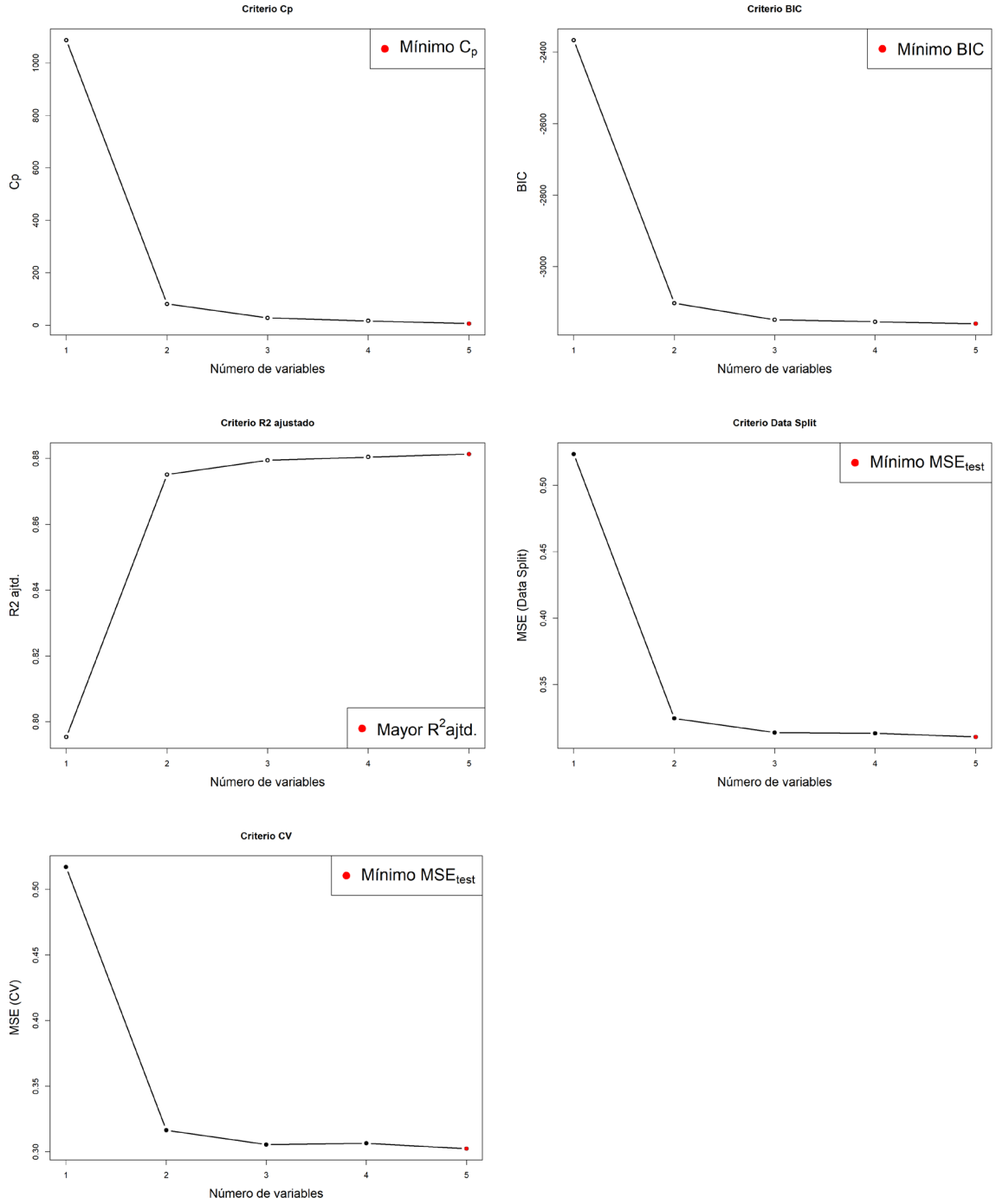

Figura 26. Modelo seleccionado para el Test 2 en función de cada criterio

Tabla 15. Muestra de 20 predicciones realizadas sobre los datos de prueba del Test 2

Rendimiento predicho	Rendimiento verdadero
1.84	2
3.38	4
3.15	3
5.00	5
3.88	4
4.43	5
5.00	5
3.17	3
0.97	1
3.55	4
3.63	4
2.88	3
5.00	5
0.54	0
0.97	1
1.80	2
2.75	3
5.00	5
1.59	1
4.16	5

8.1.3. Test 3

El conjunto de datos del Test 3 está formado por siete variables explicativas ("tiempo", "navegaciones", "s03l01", "s03l02", "s03l03", "s03l05" y "s03l06") y la variable respuesta ("aciertos"). Tal como ocurrió con los datos de los tests 1 y 2, si calculamos la matriz de correlación y la representamos gráficamente, como en la Figura 27, se puede observar que la variable "tiempo" tiene un considerable valor de correlación con las cinco variables de

revisiones. Está correlación refleja la relación que hay entre el tiempo invertido y el número de revisiones realizadas, tal como se determinó en la generación de los datos.

Entre el resto de las variables no se detecta ningún caso de correlación tan significativo. Por otro lado, tal como ha estado ocurriendo hasta ahora con los otros modelos, las variables que mejor explican la variable respuesta son las del tiempo invertido y revisiones.

Si revisamos el coeficiente VIF para cada una de las variables explicativas (Figura 28), se puede ver que la variable "tiempo" es la que mayor valor tiene seguida por las variables correspondientes a las revisiones. Esto confirma los casos de correlación observados en la matriz anterior, aunque ningún coeficiente supera la barrera de valor 5, por lo que se puede considerar que no tenemos un problema de multicolinealidad grave.

Una vez analizada la relación entre las diferentes variables, al igual que se ha hecho con los dos modelos anteriores, el siguiente paso es aplicar el método del mejor subconjunto para extraer la mejor combinación de variables para nuestro modelo de regresión lineal. En la Tabla 16 se pueden ver las mejores combinaciones obtenidas por el método en función del límite de variables establecido.

Si se tiene en cuenta una única variable se añade la variable "tiempo" por su alta correlación con la variable respuesta. En caso de tener en cuenta dos variables se añade la variable "navegaciones", ya que a diferencia de las variables de revisiones esta no tiene alta correlación con la variable "tiempo". A medida que se van teniendo en cuenta tres, cuatro, cinco, seis y siete variables explicativas se van añadiendo las variables de revisiones respectivamente.

La Figura 29 muestra el error de entrenamiento obtenido con cada uno de los modelos y, como se ha explicado en apartados anteriores, a medida que aumenta el número de variables el modelo se vuelve más flexible y este error va disminuyendo. Por lo tanto, en este caso también, el modelo con menor error de entrenamiento ha sido el modelo completo (7 variables).

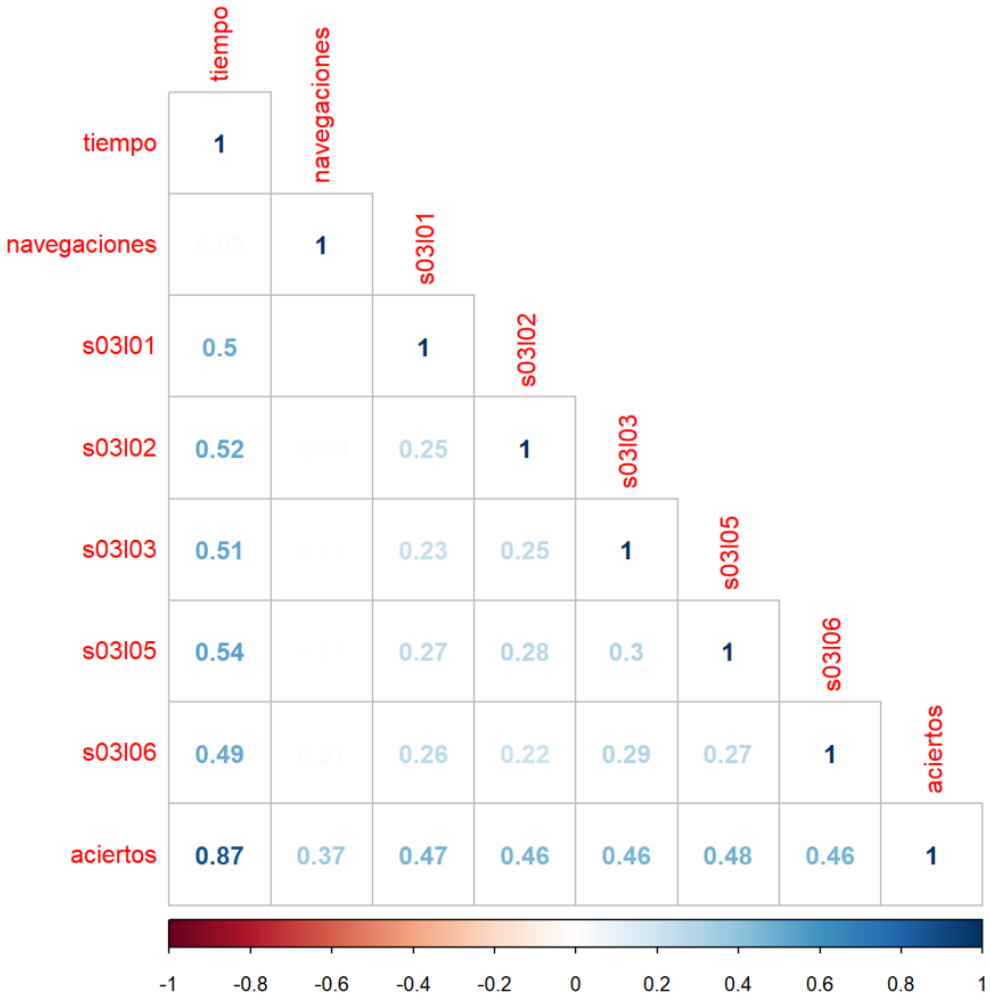

Figura 27. Correlación entre las variables del conjunto de datos del Test 3

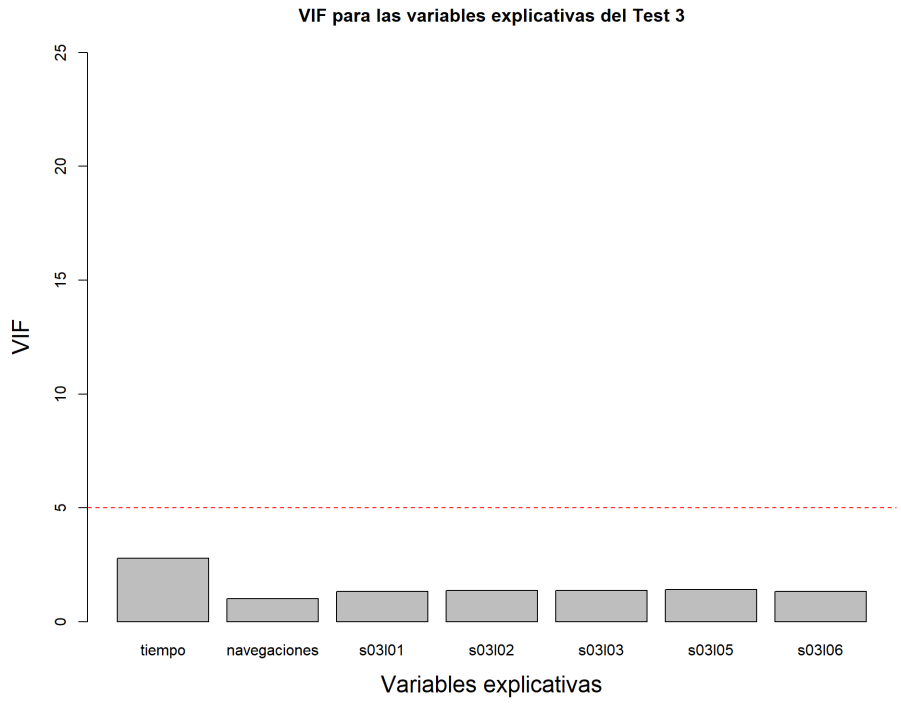

Figura 28. Factor de inflación de la varianza (VIF) para los regresores del Test 3

Tabla 16. Mejor combinación de variables del Test 3 en función del límite establecido

Nº de variables	tiempo	navegaciones	s03l01	s03l02	s03l03	s03l05	s03l06
1	x						
2	x	x					
3	x	x	x				
4	x	x	x				x
5	x	x	x		x		x
6	x	x	x	x	x		x
7	x	x	x	x	x	x	x

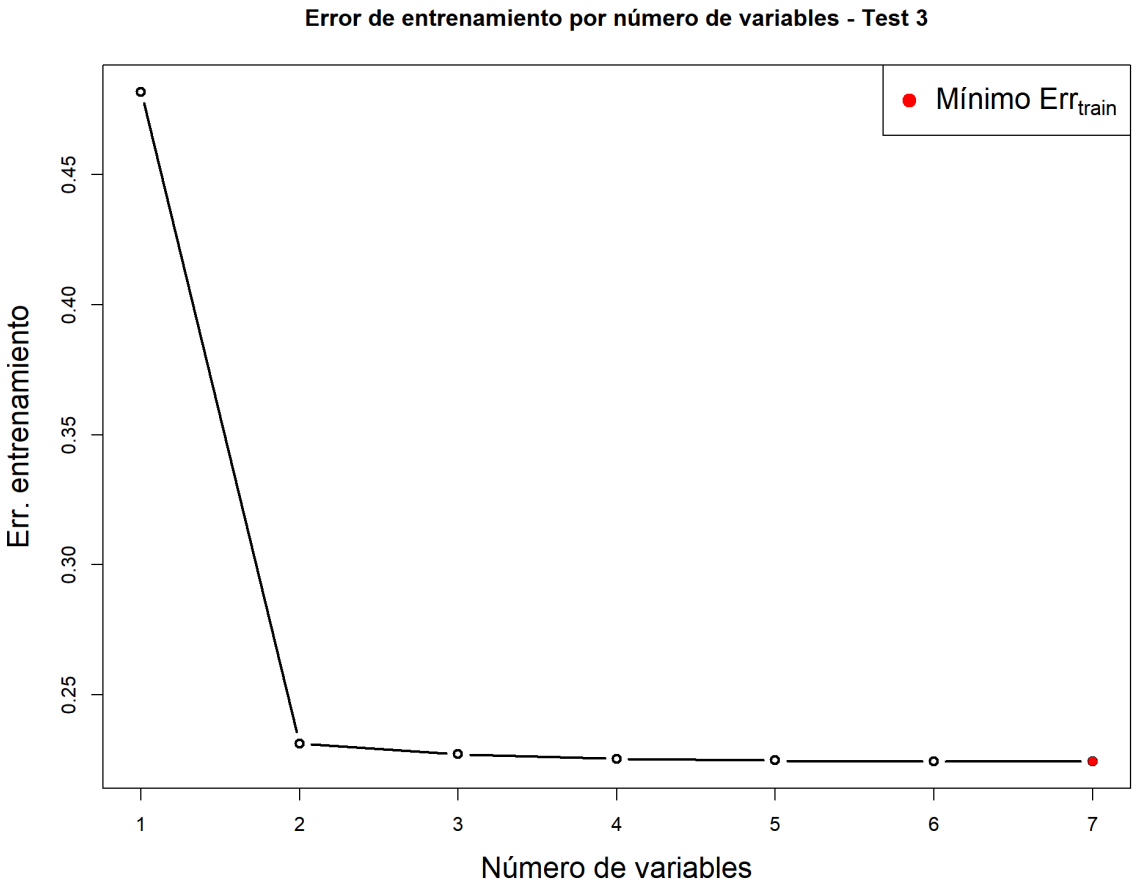

Figura 29. Error de entrenamiento de los modelos de regresión ajustados para el Test 3

En cuanto a los criterios de selección (Figura 30) hay una disparidad en la elección de los modelos. Por un lado, tres de los cinco criterios ($BIC, Data\ split$ y CV) han seleccionado el modelo de cuatro variables y, por otro lado, los otros dos criterios (C_p y R^2_{adj}) han seleccionado el modelo de seis variables.

Tabla 17. Errores de entrenamiento y generalización de los mejores modelos de regresión seleccionados para el Test 3

Núm. de variables	Error de entrenamiento	MSE
4	0.2254	0.1589
6	0.2244	0.1552

Tabla 18. Coeficientes de regresión estimados para las variables del modelo del Test 3

Term. Independiente	tiempo	navegaciones	s03l01	s03l06
0.5651	0.1479	0.1278	0.0085	0.0055

Al igual que ocurrió con el modelo del Test 1, para tomar una decisión, se va a comparar el MSE de ambos modelos estimado sobre los datos de prueba (25% de los datos). Como se puede ver en la Tabla 17, ambos modelos tienen unos errores de entrenamiento y de generalización muy similares. Por lo tanto, como es lógico, es más aconsejable escoger el modelo con menos variables por su sencillez y mayor facilidad de interpretación. La estimación de los coeficientes de regresión para cada una de las variables del modelo escogido junto al término independiente se pueden ver en la Tabla 18.

Por último, en la Tabla 19 se puede ver una muestra de 20 de las 500 predicciones llevadas a cabo sobre los datos de prueba, donde la columna de la izquierda contiene los aciertos predichos y la columna de la derecha los aciertos verdaderos del caso correspondiente. Según estos resultados, sabiendo que los aciertos posibles oscilan dentro del intervalo [0,5] y que tenemos un MSE con valor 0.1589, se puede considerar que el modelo tiene un gran rendimiento de cara a la predicción orientativa sobre los resultados que obtendrán los usuarios en el Test 3.

Criterios de selección del modelo del Test 3

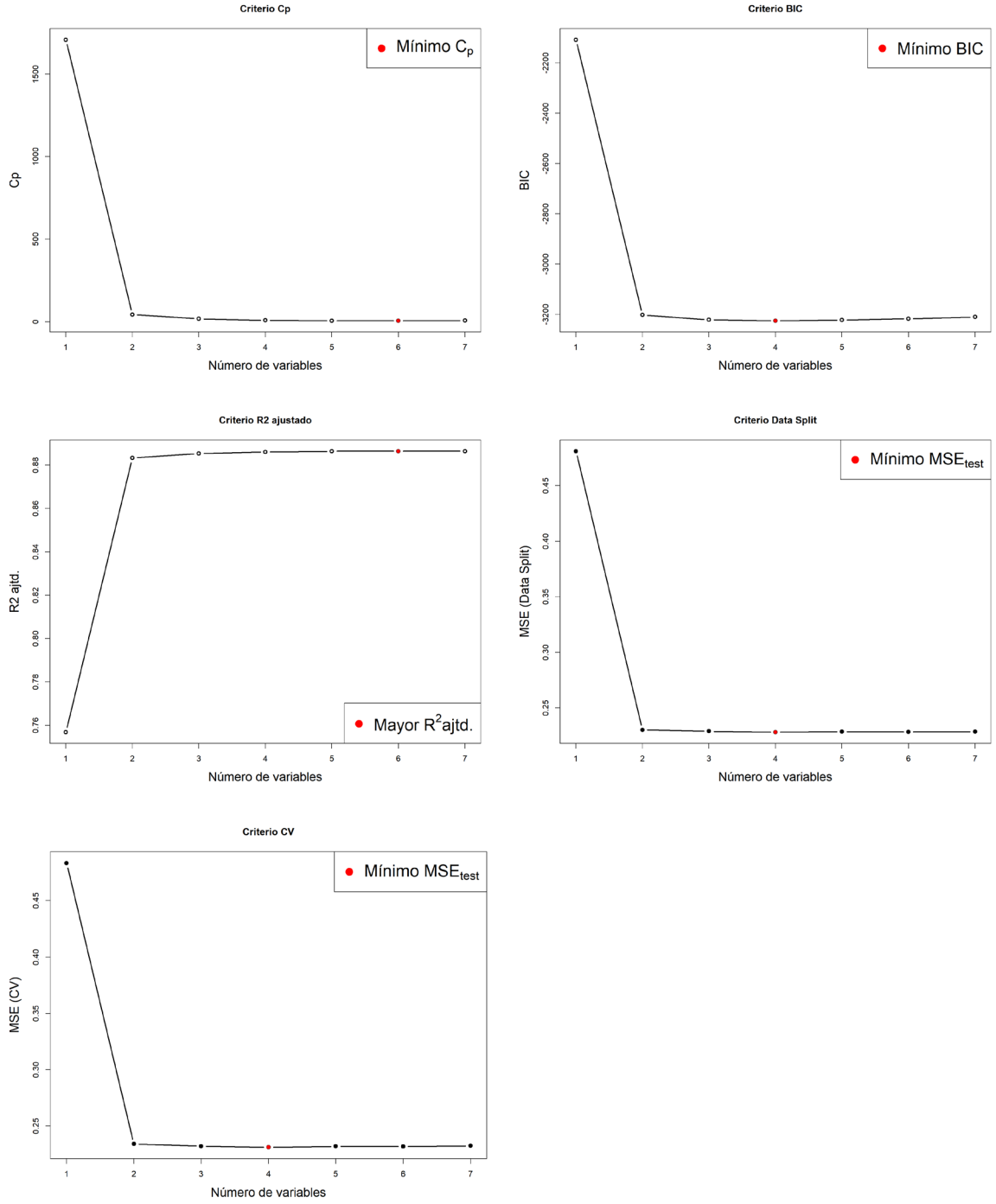

Figura 30. Modelo seleccionado para el Test 3 en función de cada criterio

Tabla 19. Muestra de 20 predicciones realizadas sobre los datos de prueba del Test 3

Rendimiento predicho	Rendimiento verdadero
3.00	3
3.05	3
4.56	5
2.31	2
3.48	4
2.82	3
2.24	2
2.52	2
2.07	2
3.22	3
2.91	3
2.94	3
4.18	5
1.96	2
1.02	1
4.06	5
3.77	4
3.51	4
4.88	5
1.18	1

8.2. Redes neuronales

Hoy en día existe una amplia variedad de arquitecturas de Redes Neuronales Artificiales o Artificial Neural Networks (ANN), cada una de ellas orientada a solucionar un tipo de problema concreto. Al igual que en los modelos de regresión lineal, entrenar redes neuronales equivale a encontrar los pesos de cada neurona de manera que a partir de unos de datos de entrada se genere la salida esperada. Es decir, consiste en la minimización del

error entre la salida de la red y la salida esperada proveniente del conjunto de observaciones de entrenamiento (función de coste) [61].

Una de las arquitecturas más básicas y utilizadas es el Perceptrón Multicapa o Multilayer Perceptron (MLP). Esta arquitectura es un tipo de red totalmente conectada o fully connected con retroalimentación hacia delante o Feed Forward Neural Network (FFNN). Está formada como mínimo por tres capas: la capa de entrada, la capa de salida y una o más capas ocultas. La capa de entrada tiene tantas neuronas como variables explicativas tenga el conjunto de datos. Por otro lado, si se va a emplear la red para resolver un problema de regresión, la capa de salida tendrá una única neurona que dará lugar al valor predicho. Sin embargo, si la red va a ser empleada para resolver un problema de clasificación, la capa de salida tendrá tantas neuronas como categorías de clasificación haya [62].

El número de capas ocultas y el número de neuronas que va a tener cada una de estas capas es un problema importante a la hora de diseñar una arquitectura ANN, ya que son parámetros que influyen de manera directa en el rendimiento de la red. Sin embargo, no hay una regla exacta para determinar estos parámetros, siendo más bien un proceso empírico que va a depender de la experiencia del autor de dicha red. En la mayoría de los casos los mejores resultados se alcanzan mediante prueba y error [63].

Los valores de las neuronas de una capa se calculan como la suma activada de las salidas ponderadas de las neuronas conectadas de la capa anterior. La activación se refiere al uso de funciones no lineales que encapsulan los pesos y las variables dando flexibilidad al modelo y en función del tipo de activación los resultados pueden estar acotados a un rango o mapeados para eliminar valores no deseados o negativos (ReLU, Soft+, Sigmoidea, Tanh, etc.) [64].

Los pesos iniciales de las neuronas de la red son inicializados aleatoriamente, pero posteriormente se van ajustando a través del proceso de propagación hacia atrás o backward propagation process. A través del proceso de entrenamiento la información se envía desde las neuronas de entrada hasta las neuronas de salida, a partir de las capas ocultas. Una vez dada la salida se calcula el error cometido en la predicción y esta señal de error se propaga hacia atrás ajustando los pesos y los sesgos de la red. Este proceso se repite para cada muestra del conjunto de datos de entrenamiento y cuando todo el conjunto de

datos ha sido consumido por la red se da por concluida una época o epoch. Normalmente suelen ser necesarias varias épocas antes de que se complete la fase de entrenamiento [65].

Para nuestra solución diseñaremos tres redes neuronales donde cada una se va a encargar de predecir el rendimiento que obtendrán los usuarios en uno de los tres tests que componen el curso e-learning. La implementación tendrá un enfoque de clasificación, es decir, si en un test se pueden obtener de 0 a 5 aciertos, se tomará cada una de estas posibilidades como una categoría y la red mediante el progreso del usuario predecirá a qué categoría pertenecerá este. De esta manera, a diferencia de los modelos de regresión, tendremos una predicción de valor entero sobre los aciertos que obtendrá el usuario.

Suponiendo que tenemos dos clases, una positiva (1) y una negativa (-1), las métricas encargadas de calcular el rendimiento de las redes neuronales en tareas de clasificación se basan en los siguientes indicadores:

- Verdaderos Positivos o True Positives (TP): representa a los individuos correctamente clasificados como de clase 1.
- Verdaderos Negativos o True Negatives (TN): representa a los individuos correctamente clasificados como de clase -1.
- Falsos positivos o False Positives (FP): representa a los individuos clasificados incorrectamente como de clase 1.
- Falsos negativos o False Negatives (FN): representa a los individuos clasificados incorrectamente como de clase -1.

En tareas de clasificación donde hay más de dos clases se toma una clase como positiva y el resto como una clase global negativa, aplicando así los mismos conceptos anteriores. Las principales métricas empleadas en las tareas de clasificación son las siguientes:

- Exactitud o Accuracy: representa la proporción de casos bien clasificados y se define como;

$$\text{Acurracy} := \frac{TP + TN}{N}$$

- Precisión o Precision: representa la tasa de acierto de las predicciones positivas, es decir, en la subpoblación de individuos clasificados como positivos (+1). Se define como;

$$\text{Precision} \coloneqq \frac{TP}{TP + FP}$$

- Sensibilidad o Recall: representa la tasa de acierto en la subpoblación de individuos que pertenecen a la clase positiva (+1). Se define como;

$$Recall \coloneqq \frac{TP}{TP + FN}$$

- F1-score: se puede utilizar para seleccionar el mejor clasificador. Se define como la media armónica entre precisión y sensibilidad;

$$F_1 \coloneqq \frac{2}{\frac{1}{sensibilidad} + \frac{1}{precisión}} = 2 \cdot \frac{precisión \cdot sensibilidad}{precisión + sensibilidad}$$

Todas estas métricas toman valores en el intervalo [0,1], y cuanto más cercano a 1 sea el valor mejor será el rendimiento del modelo evaluado.

La implementación de este apartado se va a llevar a cabo mediante el lenguaje de programación Python, haciendo uso de las funcionalidades que ofrece la librería Keras[7] para el diseño de redes neuronales.

Para nuestra solución haremos uso de una red MLP cuya arquitectura se puede ver en la Figura 31. Esta arquitectura contiene dos capas ocultas cuyo número de neuronas es fijo y tiene valor 10. En cambio, el número de neuronas de la capa de entrada (n) y el número de neuronas de la capa de salida (k) van a ser variables en función del test al que corresponda la red neuronal. Debido a que el progreso implicado en la predicción del rendimiento de cada test está formado por variables particulares, n variará en función del número de estas variables. Por otro lado, debido a que las redes van a ser entrenadas para llevar a cabo una tarea de clasificación, k variará en función del número de aciertos posibles en el test en cuestión. Concretamente para cada test estas variables tendrán los siguientes valores:

- Red neuronal del Test 1:
    - n = 4: cuatro variables explicativas (“tiempo”, “navegaciones”, “s02l01” y “s02l02”)

[7] https://keras.io/

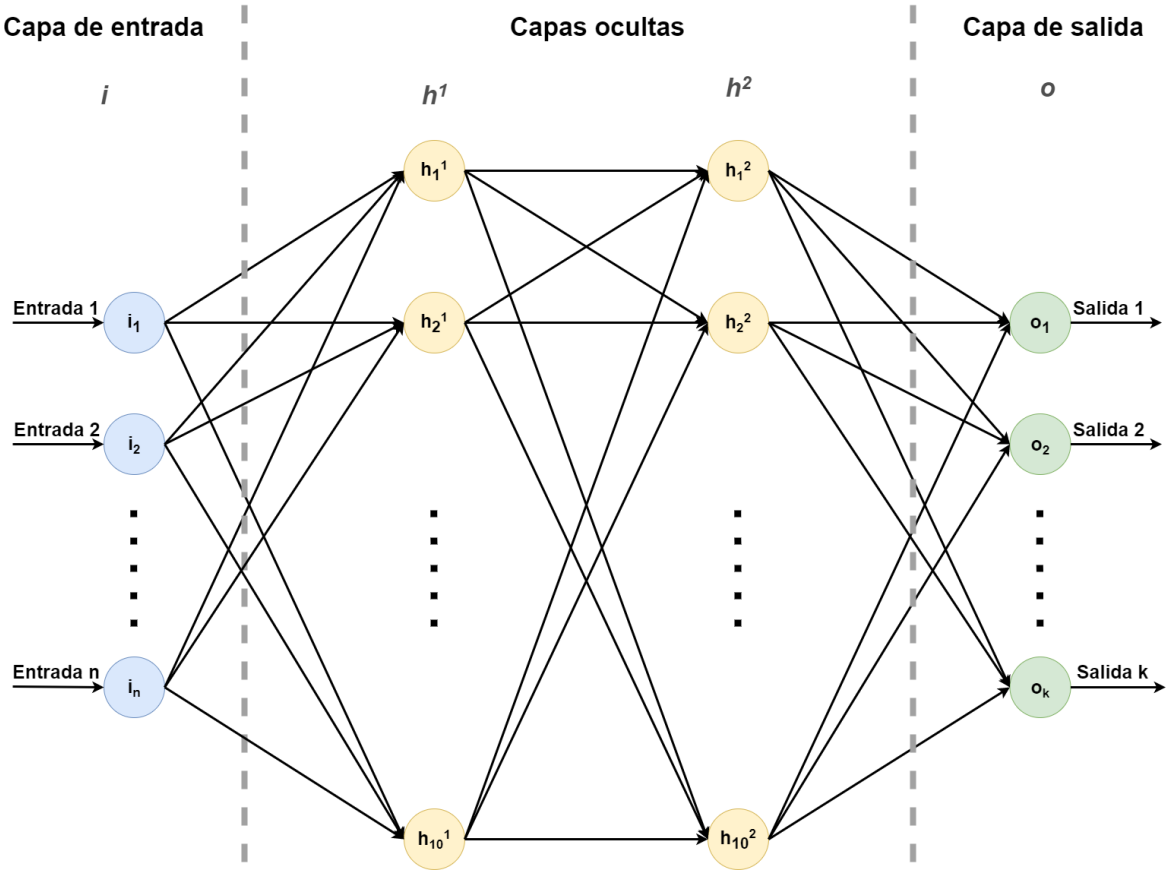

Figura 31. Diagrama de la arquitectura de la red neuronal implementada

- k = 7: el test está formado por seis cuestiones y, por lo tanto, se pueden obtener de 0 a 6 aciertos

- Red neuronal del Test 2:
  - n = 5: cinco variables explicativas ("tiempo", "navegaciones", "s02l04", "s02l05" y "s02l06")
  - k = 6: el test está formado por cinco cuestiones y, por lo tanto, se pueden obtener de 0 a 5 aciertos
- Red neuronal del Test 3:
  - n = 7: siete variables explicativas ("tiempo", "navegaciones", "s03l01", "s03l02", "s03l03", "s03l05" y "s03l06")
  - k = 6: el test está formado por cinco cuestiones y, por lo tanto, se pueden obtener de 0 a 5 aciertos

En cuanto al resto de detalles de la implementación, se ha empleado en las capas ocultas la función ReLU ($f(x) = max(0, x)$) como función de activación por su rendimiento ofrecido

Tabla 20. Métricas de clasificación de las tres redes neuronales

	Accuracy	Precision	Recall	F1-score
Test 1	0.98	0.98	0.99	0.99
Test 2	0.98	0.98	0.98	0.98
Test 3	0.98	0.98	0.99	0.98

en esta arquitectura. Por otro lado, se ha hecho uso de la función sigmoidea ($\sigma(x) = \frac{1}{1+e^{-x}}$) como función de activación de la capa de salida. Gracias a esta función de activación los valores de cada una de las neuronas de la capa de salida serán tomados como probabilidades (intervalo [0,1]) y, por lo tanto, la clase ganadora será aquella cuya probabilidad sea más alta. Por último, al igual que en los modelos de regresión lineal, se ha reservado el 25% de los datos para emplearlo como conjunto de datos de prueba. El proceso de entrenamiento se ha llevado a cabo durante 500 épocas o epochs, con el MSE como función de coste y con el método Adam como algoritmo de optimización.

Ante conjuntos de datos donde las clases o categorías están muy desequilibradas o la proporción de individuos que pertenece a una cierta clase es próxima a 1, la tasa de acierto o accuracy no es una medida de rendimiento adecuada. Por ello, existen medidas complementarias como la sensibilidad, precisión y F1-score. En nuestro caso, para asegurarnos de medir con certeza el rendimiento de cada red neuronal, vamos a calcular estas cuatro métricas.

Tras hacer la predicción sobre los datos de prueba de cada uno de los modelos, se han calculado los valores de las métricas anteriormente mencionadas y cuyos valores se pueden ver en la Tabla 20. En esta tabla cada columna representa a cada una de las métricas y las filas representan el test al cual pertenece el modelo evaluado. Como se puede ver, los resultados de clasificación para cada modelo son excelentes por consenso entre las cuatro métricas. Por otro lado, en la Tabla 21, se ha representado para cada test el contraste entre el rendimiento predicho y el rendimiento verdadero mediante una muestra de las primeras 20 predicciones realizadas. Concretamente, del total de las predicciones, se acierta el 98.40% de los casos predichos para el Test 1, el 97.60% de casos predichos para el Test 2 y el 98.20% de casos predichos para el Test 3.

Tabla 21. Muestra de 20 predicciones realizadas por las tres redes neuronales

Aciertos predichos			Aciertos verdaderos		
Test 1	**Test 2**	**Test 3**	**Test 1**	**Test 2**	**Test 3**
6	2	3	6	2	3
0	4	3	0	4	3
5	3	5	5	3	5
5	5	2	5	5	2
6	4	3	6	4	4
4	5	3	4	5	3
4	5	2	4	5	2
6	3	2	6	3	2
2	1	2	2	1	2
2	4	3	2	4	3
4	4	3	4	4	3
1	3	3	2	3	3
2	5	5	2	5	5
5	0	2	5	0	2
0	1	1	0	1	1
4	2	5	4	2	5
5	3	4	5	3	4
6	5	4	6	5	4
4	1	5	4	1	5
6	5	1	6	5	1

9. Resultados

Tras completar los apartados anteriores, tenemos como resultado dos posibles soluciones a nuestro objetivo principal, el cual consistía en implementar un sistema capaz de predecir el rendimiento que van a tener los usuarios en las diferentes pruebas o test de un curso e-learning a partir del progreso monitorizado de estos.

Para lograr este objetivo se ha abordado la solución desde dos enfoques; un enfoque de regresión y otro enfoque de clasificación. Se comenzó buscando la solución a través del enfoque de regresión, mediante el cual se ha conseguido implementar un modelo predictivo para cada uno de los tres tests que componen el curso. El rendimiento de estos modelos ha sido muy bueno, obteniendo unos valores de MSE muy bajos dentro de la escala de los aciertos posibles de cada test.

Si bien, los modelos anteriores permiten obtener una aproximación muy buena acerca del rendimiento que pueden llegar a obtener los usuarios en los tests, cuentan con el inconveniente de que predicen valores continuos. Por ello, se buscó la misma solución bajo un enfoque de clasificación de manera que los resultados predichos fueran valores discretos. Para ello se han implementado tres redes neuronales bajo una arquitectura MLP encargadas de la predicción del rendimiento de cada uno de los tests del curso e-learning. La salida de estas redes neuronales es un vector de probabilidades con tantos índices como aciertos posibles tenga el test en cuestión, y el índice con mayor probabilidad corresponde a la clase de clasificación final. El rendimiento obtenido con este enfoque también ha sido muy bueno, con unos porcentaje de acierto alrededor del 98% en las tres redes neuronales.

Por lo tanto, el segundo enfoque se puede considerar la solución resultante final por su capacidad de predicción discreta, la cual permite una interpretación más fácil y directa para los usuarios. Sin embargo, la solución obtenida con el primer enfoque no es desechable. De hecho, los modelos de regresión pueden ser empleados como soporte a los resultados de las redes neuronales, aportando mayor seguridad en la predicción si los resultados son similares o como advertencia de una predicción errónea en caso de una alta disparidad.

Tabla 22. Rendimiento global de las soluciones implementadas

	Test 1	**Test 2**	**Test 3**
MSE (Regresión)	0.2177	0.1524	0.1589
Accuracy % (Clasificación)	98.40	97.60	98.20

10. Conclusiones y trabajo futuro

Durante este proyecto se ha pasado por diferentes fases de trabajo mediante las cuales se han ido cumpliendo los subobjetivos que han permitido lograr el objetivo principal. El punto de partida fueron los datos provenientes de la monitorización de la actividad de los usuarios dentro de un curso e-learning. Para dichos datos se ha creado un flujo automatizado de preprocesamiento que genera la estructura de datos final, adecuada para trabajar con los modelos predictivos. Por otro lado, debido a la insuficiente cantidad de datos, se ha llevado a cabo un detallado proceso de generación de datos mediante el cual se ha podido aumentar la cantidad disponible de estos. Finalmente, con los datos listos, se han implementado los modelos predictivos basándonos en diferentes estrategias y métodos para dar con la mejor solución para nuestro problema.

El sistema predictivo final desarrollado ha demostrado tener una gran capacidad sobre la predicción del rendimiento de los usuarios en el e-learning en base al progreso de estos. Esto tiene un importante impacto en la calidad de la docencia del e-learning, permitiendo orientar a los usuarios de cara a sus próximas pruebas de evaluación mediante el conocimiento con antelación de su nivel de preparación para estas. Y, por otro lado, permite a los instructores o docentes conocer con antelación el rendimiento que pueden llegar a tener sus alumnos. Esto les capacita para poder proporcionar en una etapa temprana un soporte personalizado a los usuarios que lo necesiten, evitando así casos de abandono por causa de posibles barreras de aprendizaje a las que se pueden enfrentar determinados usuarios.

Por último, se plantean las siguientes líneas de trabajo futuro que permitirían continuar con el crecimiento de este proyecto:

- Poner en producción el sistema de predicción. Como se ha explicado en los primeros apartados de este trabajo, partimos de los datos monitorizados en el curso e-learning desarrollado en el TFG. Este curso se encuentra publicado y disponible en una plataforma web, por lo que el siguiente paso sería añadir a dicha aplicación web la funcionalidad de predicción del rendimiento de los usuarios a partir de la solución desarrollada en este trabajo.

- Atraer a una mayor masa de usuarios para recopilar una mayor cantidad de datos reales. Si bien, los datos sintéticos que han sido generados para desarrollar la solución representan muy bien las características reales del entorno del curso e-learning, sería ideal poder alcanzar la misma cantidad de datos en base al progreso de usuarios reales. Para ello habría que trabajar en una mayor promoción del curso, planteando la posibilidad de ofrecer a los usuarios algún certificado por parte de alguna entidad colaboradora que sea reconocida en el sector.
- Predecir el "engagement" o riesgo de abandono de los estudiantes. Como medida de apoyo a la predicción del rendimiento, el indicador de riesgo de abandono sería de gran ayuda de cara a poder detectar estudiantes que se encuentran ante este riesgo o ante un estancamiento en el curso e-learning por causa de posibles barreras de aprendizaje. Esto permitiría a los docentes, junto a la predicción temprana del rendimiento, localizar con mayor precisión los grupos de estudiantes que necesitan una asistencia personalizada.

Referencias

[1] R. E. Derouin, B. A. Fritzsche and E. Salas, "E-Learning in Organizations", *Journal of Management*, vol. 31, no. 6, pp. 920–940, dic. 2005, doi: 10.1177/0149206305279815.

[2] V. Chang, "Review and discussion: E-learning for academia and industry", *International Journal of Information Management*, vol. 36, no. 3, pp. 476–485, jun. 2016, doi: 10.1016/J.IJINFOMGT.2015.12.007.

[3] "1.37 billion students now home as COVID-19 school closures expand, ministers scale up multimedia approaches to ensure learning continuity", *UNESCO*, mzo. 2020. Accedido: ag. 01, 2021 [En línea]. Disponible: https://en.unesco.org/news/137-billion-students-now-home-covid-19-school-closures-expand-ministers-scale-multimedia

[4] "COVID-19 : with half of world's student population out of school, UNESCO launches coalition to accelerate remote learning solutions", *UNESCO*, mzo. 2020. Accedido: ag. 01, 2021 [En línea]. Disponible: https://en.unesco.org/news/covid-19-half-worlds-student-population-out-school-unesco-launches-coalition-accelerate-remote

[5] A. López, "El aprendizaje en línea en 2021: de la urgencia a la calidad", *UOC - Universitat Oberta de Catalunya*, dic. 2020. Accedido: ag. 01, 2021 [En línea]. Disponible: https://www.uoc.edu/portal/es/news/actualitat/2020/463-elearning-2021-calidad.html

[6] V. D. Soni, "Global Impact of E-learning during COVID 19", *SSRN Electronic Journal*, jun. 2020, doi: 10.2139/SSRN.3630073.

[7] M. Ebner, S. Schön, C. Braun, M. Ebner, Y. Grigoriadis, M. Hass, P. Leitner and B. Taraghi, "COVID-19 Epidemic as E-Learning Boost? Chronological Development and Effects at an Austrian University against the Background of the Concept of 'E-Learning Readiness'", *Future Internet*, vol. 12, no. 6, p. 94, my. 2020, doi: 10.3390/FI12060094.

[8] A. Y. Alqahtani and A. A. Rajkhan, "E-Learning Critical Success Factors during the COVID-19 Pandemic: A Comprehensive Analysis of E-Learning Managerial Perspectives", *Education Sciences*, vol. 10, no. 9, p. 216, ag. 2020, doi: 10.3390/EDUCSCI10090216.

[9] S. Abbasi, T. Ayoob, A. Malik and S. I. Memon, "Perceptions of students regarding E-learning during Covid-19 at a private medical college", *Pakistan Journal of Medical Sciences*, vol. 36, no. COVID19-S4, p. S57, my. 2020, doi: 10.12669/PJMS.36.COVID19-S4.2766.

[10] E. Aboagye, J. A. Yawson and K. N. Appiah, "COVID-19 and E-Learning: the Challenges of Students in Tertiary Institutions", *Social Education Research*, pp. 1–8, jun. 2020, doi: 10.37256/SER.212021422.

[11] M. Amin Almaiah, A. Al-Khasawneh and A. Althunibat, "Exploring the critical challenges and factors influencing the E-learning system usage during COVID-19 pandemic", *Education and Information Technologies*, vol. 25, pp. 5261–5280, my. 2020, doi: 10.1007/s10639-020-10219-y.

[12] O. Moukket Rkizat, "Estudio de la actividad de los usuarios en el e-learning que permita identificar barreras de aprendizaje para las personas con discapacidad", jun. 2020. Accedido: ag. 01, 2021 [En línea]. Disponible: http://rua.ua.es/dspace/handle/10045/107798

[13] N. Karlovcec, N. Karlovcec, T. Skala and S. Saina, "Computer Science Education: Differences Between E-learning and Classical Approach", *E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, vol. 2005, no. 1, pp. 800–805, oct. 2005. Accedido: ag. 01, 2021 [En línea]. Disponible: https://www.learntechlib.org/primary/p/21277/

[14] B. Holder, "An investigation of hope, academics, environment, and motivation as predictors of persistence in higher education online programs", *The Internet and Higher Education*, vol. 10, no. 4, pp. 245–260, en. 2007, doi: 10.1016/J.IHEDUC.2007.08.002.

[15] F. Ullah, J. Wang, M. Farhan, S. Jabbar, Z. Wu and S. Khalid, "Plagiarism detection in students' programming assignments based on semantics: multimedia e-learning based smart assessment methodology", *Multimedia Tools and Applications*, vol. 79, no. 13, pp. 8581–8598, mzo. 2018, doi: 10.1007/S11042-018-5827-6.

[16] A. Moubayed, M. Injadat, A. B. Nassif, H. Lutfiyya and A. Shami, "E-Learning: Challenges and Research Opportunities Using Machine Learning Data Analytics", *IEEE Access*, vol. 6, pp. 39117–39138, jul. 2018, doi: 10.1109/ACCESS.2018.2851790.

[17] S. Farid, R. Ahmad, I. A. Niaz, M. Arif, S. Shamshirband and M. D. Khattak, "Identification and prioritization of critical issues for the promotion of e-learning in Pakistan", *Computers in Human Behavior*, vol. 51, pp. 161–171, oct. 2015, doi: 10.1016/J.CHB.2015.04.037.

[18] L. B. Krithika and G. G. Lakshmi Priya, "Student Emotion Recognition System (SERS) for e-learning Improvement Based on Learner Concentration Metric", *Procedia Computer Science*, vol. 85, pp. 767–776, en. 2016, doi: 10.1016/J.PROCS.2016.05.264.

[19] S. Marshall, "A Quality Framework for Continuous Improvement of e-Learning: The e-Learning Maturity Model", *Journal of Distance Education*, vol. 24, no. 1, pp. 143-166, 2010. Accedido: ag. 01, 2021 [En línea]. Disponible: https://eric.ed.gov/?id=EJ892382

[20] S. B. Aher and L. M. R. J. Lobo, "Combination of machine learning algorithms for recommendation of courses in E-Learning System based on historical data", *Knowledge-Based Systems*, vol. 51, pp. 1–14, oct. 2013, doi: 10.1016/J.KNOSYS.2013.04.015.

[21] S. S. Khanal, P. W. C. Prasad, A. Alsadoon and A. Maag, "A systematic review: machine learning based recommendation systems for e-learning", *Education and Information Technologies*, vol. 25, no. 4, pp. 2635–2664, jul. 2020, doi: 10.1007/S10639-019-10063-9.

[22] M. A. P. Junemann, P. A. S. Lagos and R. C. Arriagada, "Neural Networks to Predict Schooling Failure/Success", *Lecture Notes in Computer Science (including subseries*

Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 4528, no. 2, pp. 571–579, 2007, doi: 10.1007/978-3-540-73055-2_59.

[23] T. Wang and A. Mitrovic, "Using neural networks to predict student's performance", *International Conference on Computers in Education*, pp. 969–973, dic. 2002, doi: 10.1109/CIE.2002.1186127.

[24] A. Cripps, "Using artificial neural nets to predict academic performance", *Proceedings of the ACM Symposium on Applied Computing*, febr. 1996, pp. 33–37, doi: 10.1145/331119.331137.

[25] D. Buenaño-Fernández, D. Gil and S. Luján-Mora, "Application of Machine Learning in Predicting Performance for Computer Engineering Students: A Case Study", *Sustainability*, vol. 11, no. 10, p. 2833, my. 2019, doi: 10.3390/SU11102833.

[26] O. Moscoso-Zea, P. Saa and S. Luján-Mora, "Evaluation of algorithms to predict graduation rate in higher education institutions by applying educational data mining", *Australasian Journal of Engineering Education*, vol. 24, no. 1, pp. 4–13, en. 2019, doi: 10.1080/22054952.2019.1601063.

[27] S. J. Sheel, D. Vrooman, R. S. Renner and S. K. Dawsey, "A Comparison of Neural Networks and Classical Discriminant Analysis in Predicting Students' Mathematics Placement Examination Scores", *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2074, pp. 952–957, jul. 2001, doi: 10.1007/3-540-45718-6_100.

[28] D. Kalles and C. Pierrakeas, "Analyzing student performance in distance learning with genetic algorithms and decision trees", *Applied Artificial Intelligence*, vol. 20, no. 8, pp. 655–674, oct. 2006, doi: 10.1080/08839510600844946.

[29] S. Kotsiantis, C. Pierrakeas and P. Pintelas, "Predicting students' performance in distance learning using machine learning techniques", *Applied Artificial Intelligence*, vol. 18, no. 5, pp. 411–426, my. 2004, doi: 10.1080/08839510490442058.

[30] T. M. Haladyna, "Developing and validating multiple-choice test ítems", 3rd ed. *Routledge*, nov. 2015.

[31] I. Lykourentzou, I. Giannoukos, G. Mpardis, V. Nikolopoulos and V. Loumos, "Early and Dynamic Student Achievement Prediction in E-Learning Courses Using Neural Networks", *Wiley Online Library*, vol. 60, no. 2, pp. 372–380, febr. 2008, doi: 10.1002/asi.20970.

[32] J. F. Feldhusen and S. M. Moon, "Grouping Gifted Students: Issues and Concerns", *Gifted Child Quarterly*, vol. 36, no. 2, pp. 63–67, abr. 1992, doi: 10.1177/001698629203600202.

[33] B. Oakley, R. Brent, R. M. Felder and I. Elhajj, "Turning student groups into effective teams", *Journal of Student Centered Learning*, vol. 2, no. 1, pp. 9-34, en. 2004, doi: 10.1.1.422.8179.

[34] S. Carr, "As Distance Education Comes of Age, the Challenge Is Keeping the Students", *The Chronicle of Higher Education*, febr. 2000. Accedido: ag. 01, 2021 [En línea]. Disponible: https://www.chronicle.com/article/as-distance-education-comes-of-age-the-challenge-is-keeping-the-students/

[35] M. Xenos, "Prediction and assessment of student behaviour in open and distance education in computers using Bayesian networks", *Computers and Education*, vol. 43, no. 4, pp. 345–359, dic. 2004, doi: 10.1016/J.COMPEDU.2003.09.005.

[36] K. Frankola, "Why Online Learners Drop Out", *Workforce Management Software,* jun. 2001. Accedido: ag. 01, 2021 [En línea]. Disponible: https://www.workforce.com/uk/news/why-online-learners-drop-out

[37] W. Doherty, "An analysis of multiple factors affecting retention in Web-based community college courses", *Internet and Higher Education*, vol. 9, no. 4, pp. 245–255, dic. 2006, doi: 10.1016/J.IHEDUC.2006.08.004.

[38] V. Carter, "Do media influence learning? Revisiting the debate in the context of distance education", *Open Learning*, vol. 11, no. 1, pp. 31–40, 1996, doi: 10.1080/0268051960110104.

[39] K. Blom and D. Meyers, "Quality indicators in vocational education and training: international perspectives", *National Centre for Vocational Education Research*, oct.

2003. Accedido: ag. 01, 2021 [En línea]. Disponible: https://www.ncver.edu.au/research-and-statistics/publications/all-publications/quality-indicators-in-vocational-education-and-training-international-perspectives

[40] N. Mduma, K. Kalegele and D. Machuve, "A survey of machine learning approaches and techniques for student dropout prediction", *Data Science Journal*, vol. 18, no. 1, abr. 2019, doi: 10.5334/DSJ-2019-014.

[41] M. Hussain, W. Zhu, W. Zhang and S. M. R. Abidi, "Student Engagement Predictions in an e-Learning System and Their Impact on Student Course Assessment Scores", *Computational Intelligence and Neuroscience*, vol. 2018, oct. 2018, doi: 10.1155/2018/6347186.

[42] C. Márquez-Vera, A. Cano, C. Romero, A. Y. M. Noaman, H. M. Fardoun and S. Ventura, "Early dropout prediction using data mining: a case study with high school students", *Expert Systems*, vol. 33, no. 1, pp. 107–124, febr. 2016, doi: 10.1111/EXSY.12135.

[43] N. Nistor and K. Neubauer, "From participation to dropout: Quantitative participation patterns in online university courses", *Computers & Education*, vol. 55, no. 2, pp. 663–672, sept. 2010, doi: 10.1016/J.COMPEDU.2010.02.026.

[44] C. Burgos, M. L. Campanario, D. de la Peña, J. A. Lara, D. Lizcano and M. A. Martínez, "Data mining for modeling students' performance: A tutoring action plan to prevent academic dropout", *Computers & Electrical Engineering*, vol. 66, pp. 541–556, febr. 2018, doi: 10.1016/J.COMPELECENG.2017.03.005.

[45] M. D. Roblyer, L. Davis, S. C. Mills, J. Marshall and L. Pape, "Toward Practical Procedures for Predicting and Promoting Success in Virtual School Students", *American Journal of Distance Education*, vol. 22, no. 2, pp. 90–109, my. 2008, doi: 10.1080/08923640802039040.

[46] M. Xenos, C. Pierrakeas and P. Pintelas, "A survey on student dropout rates and dropout causes concerning the students in the Course of Informatics of the Hellenic

Open University", *Computers & Education*, vol. 39, no. 4, pp. 361–377, dic. 2002, doi: 10.1016/S0360-1315(02)00072-6.

[47] J. W. Vare, W. Mark, E. R. Dewalt and E. D. Dockery, "Predicting Student Retention in Teacher Education Programs", febr. 2000. Accedido: ag. 01, 2021 [En línea]. Disponible: https://eric.ed.gov/?id=ED440072

[48] G. Zhang, T. J. Anderson, M. W. Ohland and B. R. Thorndyke, "Identifying factors influencing engineering student graduation: A longitudinal and cross-institutional study", *Journal of Engineering Education*, vol. 93, no. 4, pp. 313–320, oct. 2004, doi: 10.1002/J.2168-9830.2004.TB00820.X.

[49] C. C. Newell, "Learner characteristics as predictors of online course completion among nontraditional technical college students", *University of Georgia*, 2007. Accedido: ag. 01, 2021 [En línea]. Disponible: https://esploro.libs.uga.edu/esploro/outputs/9949333248802959

[50] C. V. Ukpabi, "Formulating a Prediction Model of Retention Rate in the University of North Carolina System", abr. 2005, Accedido: ag. 01, 2021 [En línea]. Disponible: https://repository.lib.ncsu.edu/handle/1840.16/3014

[51] S. B. Kotsiantis, C. J. Pierrakeas and P. E. Pintelas, "Preventing Student Dropout in Distance Learning Using Machine Learning Techniques", *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, vol. 2774, no. 2, pp. 267–274, 2003, doi: 10.1007/978-3-540-45226-3_37.

[52] S. Herzog, "Estimating student retention and degree-completion time: Decision trees and neural networks vis-à-vis regression", *New Directions for Institutional Research*, vol. 2006, no. 131, pp. 17–33, sept. 2006, doi: 10.1002/IR.185.

[53] L. G. Moseley and D. M. Mead, "Predicting who will drop out of nursing courses: A machine learning exercise", *Nurse Education Today*, vol. 28, no. 4, pp. 469–475, my. 2008, doi: 10.1016/J.NEDT.2007.07.012.

[54] M. Xenos, "Prediction and assessment of student behaviour in open and distance education in computers using Bayesian networks", *Computers and Education*, vol. 43, no. 4, pp. 345–359, dic. 2004, doi: 10.1016/J.COMPEDU.2003.09.005.

[55] D. Buenaño-Fernandez, S. Luján-Mora and D. Gil, "A Hybrid Machine Learning Approach for the Prediction of Grades in Computer Engineering Students", *Springer Proceedings in Complexity*, pp. 125–134, abr. 2019, doi: 10.1007/978-3-030-30809-4_13.

[56] I. Lykourentzou, I. Giannoukos, V. Nikolopoulos, G. Mpardis and V. Loumos, "Dropout prediction in e-learning courses through the combination of machine learning techniques", *Computers & Education*, vol. 53, no. 3, pp. 950–965, nov. 2009, doi: 10.1016/J.COMPEDU.2009.05.010.

[57] P. McCullagh and J. A. Nelder, "Generalized Linear Models", *Regression Analysis with Application G.B. Wetherill*, no. 2, p. 28, en. 2019, doi: 10.1201/9780203753736.

[58] G. Hutcheson, "Ordinary Least-Squares Regression", *The Multivariate Social Scientist*, pp. 56–113, jul. 2011, doi: 10.4135/9780857028075.D49.

[59] A. Alin, "Multicollinearity", *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 3, pp. 370–374, my. 2010, doi: 10.1002/WICS.84.

[60] G. Chandrashekar and F. Sahin, "A survey on feature selection methods", *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, en. 2014, doi: 10.1016/J.COMPELECENG.2013.11.024.

[61] H. Ramchoun, M. Amine and J. Idrissi, "Multilayer Perceptron: Architecture Optimization and Training multi-criteria learning and nonlinear optimization View project", *Article in International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 4, pp. 1–26, en. 2016, doi: 10.9781/ijimai.2016.415.

[62] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning", *Nature*, vol. 521, pp. 436–444, my. 2015, doi: 10.1038/nature14539.

[63] M. M. Saritas and A. Yasar, "Performance Analysis of ANN and Naive Bayes Classification Algorithm for Data Classification", *International Journal of Intelligent*

Systems and Applications in Engineering, vol. 7, no. 2, pp. 88–91, jun. 2019, doi: 10.18201//IJISAE.2019252786.

[64] Z. Car, S. Baressi Šegota, N. Anđelić, I. Lorencin and V. Mrzljak, "Modeling the Spread of COVID-19 Infection Using a Multilayer Perceptron", *Computational and Mathematical Methods in Medicine*, vol. 2020, my. 2020, doi: 10.1155/2020/5714714.

[65] M. Cilimkovic, "Back Propagation Algorithm Neural Networks and Back Propagation Algorithm", *Institute of Technology Blanchardstown*, 2015.